HOBBY: NO BREAK PARA O SEU TK

A REVISTA DOS USUÁRIOS DO TK

MICROHOBBY

ANO I – N.º 7 – 1984

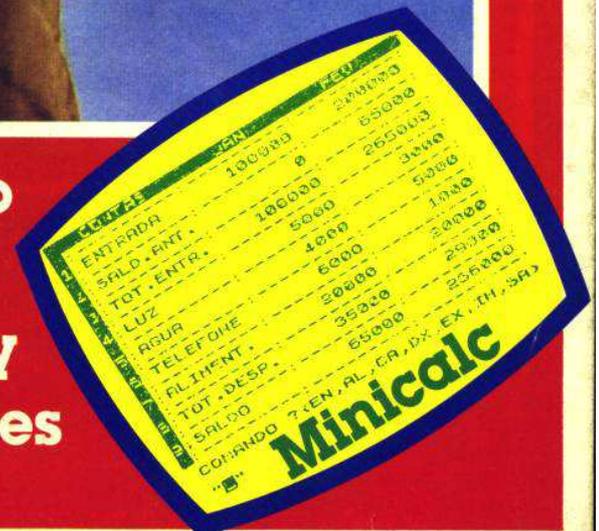
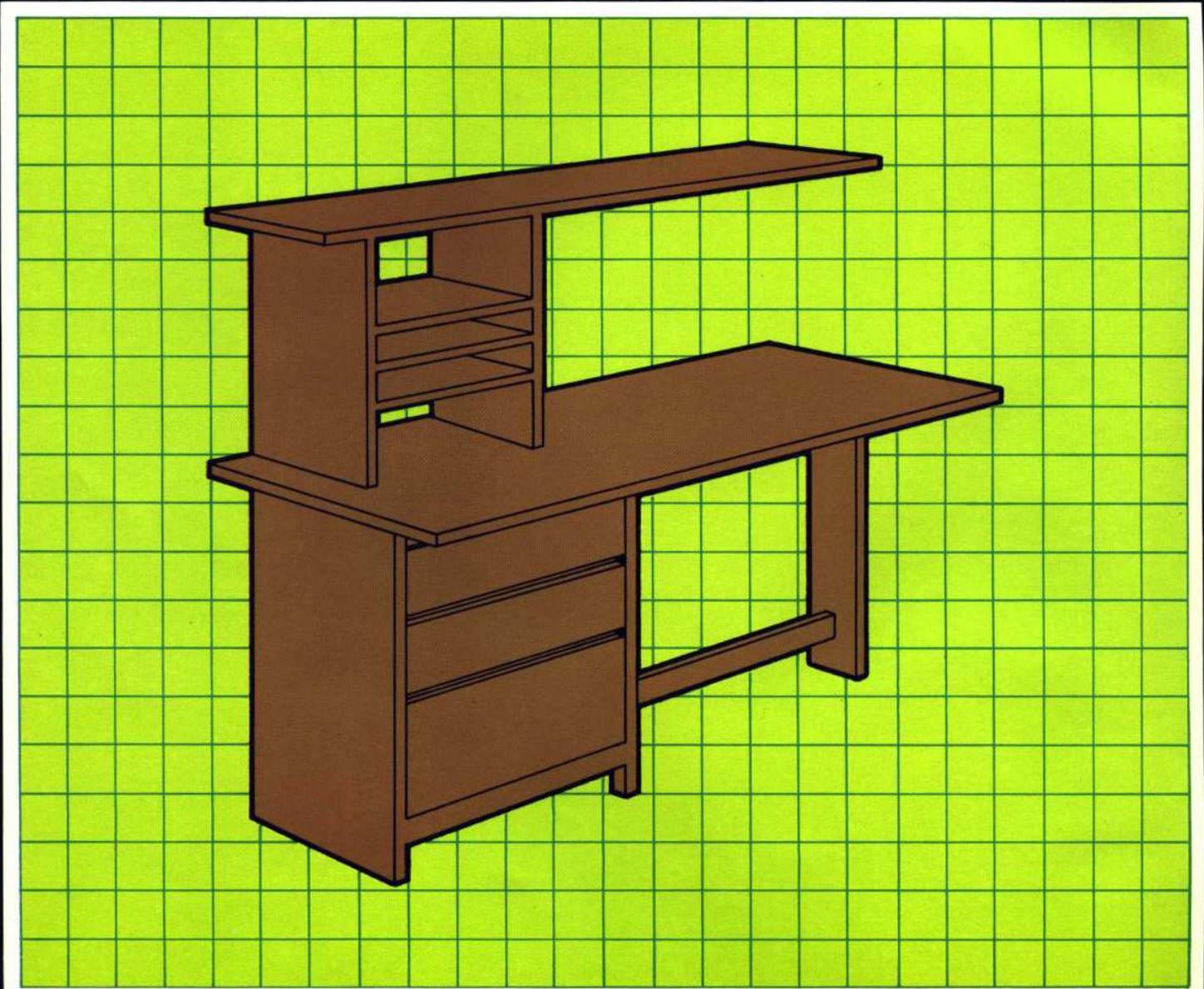
Cr$ 1.400,00

DUELO 2 kBytes

14 Programas Para Meditação

Fita do Mês: Compilador

Cursos de Basic e Assembly

Quebra-Cabeça: As Sete Pontes

Minicalc

Expediente

DIRETOR EDITORIAL
Pierluigi Piazzi
JORNALISTA RESPONSÁVEL
Aristides Ribas Fº
EDITOR
Álvaro A.L. Domingues
COORDENAÇÃO EDITORIAL
Ana Lúcia de Alcântara
ASSESSORIA TÉCNICA
Flavio Rossini
José Wilson Tucci
ANÁLISE E REDAÇÃO:
Nancy Mitie Ariga
Carlos Eduardo Rocha Salvato
Gustavo Egidio de Almeida
Renato da Silva Oliveira
Roberto Bertini Renzetti
ARTE
Cassiano Roda
Eliana Santos Queiroz
Fatima M. Rossini Gouveia
Osmère Sarkis
PRODUÇÃO GRÁFICA
José Carlos Sarkis
COLABORADORES
Igor Sartori, Nilson D. Martello, Pedro Antonio C.P. de Souza, Tanios Hamzo
CORRESPONDENTES:
Londres – Robert L. Lloyd
Paris – Alain Richard
New York -- Natan Portnoy
Milão – Bruno Origo
PUBLICIDADE:
Aurio J. Mosolino
Edson R. Silva
André Antiqueira Filho
Elias Oliveira Gonçalves
CIRCULAÇÃO E ASSINATURAS:
Atílio Debatin
MICROHOBBY é editada mensalmente por **MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO LTDA, INPI 2992** Livro A.
Endereço para correspondência:
Rua Bahia, 1049 – Cx. Postal 54121 – CEP 01296 – São Paulo, SP –
Para solicitar assinaturas (12 números), enviar cheque nominal cruzado à **MICROMEGA P.M.D. LTDA.**, no valor de Cr$ 14.800,00.
Tiragem desta Edição: 30.000 exemplares
NÚMERO 7

FOTOLITOS
Ponto Reproduções Gráficas SC Ltda.
Fone: 814-6311

Índice

Editorial

Muitos insensatos tentam amenizar as profecias apocalípticas dos cientistas falando em guerra nuclear limitada, em chances de sobrevivência e outras falsas esperanças. Um pouco de conhecimento de ecologia e meteorologia, porém, permite calcular as consequências de um conflito nuclear: as chances de sobrevivência na Terra são reduzidíssimas e ainda assim para algumas espécies, **entre as quais, com certeza, não se inclui a humana.**

O que fazer? É hora de todos os indivíduos perceberem que o conflito básico não é político ou ideológico: o problema não é mais de esquerda contra direita ou moderados contra radicais. O problema básico é de indivíduos contra o totalitarismo mais ou menos acentuado de **todos** *os Estados. Enquanto o indivíduo não for respeitado em seus direitos mais básicos, a humanidade como um todo estará seriamente ameaçada.*

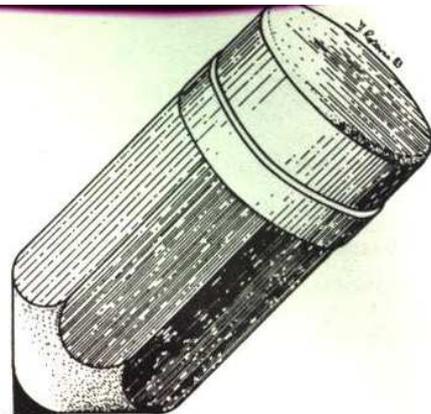
Esta ameaça se torna mais concreta no momento em que as nações formam sociedade cada vez mais informatizadas, acentuando o poder de controle daquela meia dúzia de paranóicos sobre todos os cidadãos. **1984** *está aí e o* **Big Brother** *já é uma realidade. No Brasil, talvez se chame SNI, na URSS, KGB, mas ele, sob múltiplas facetas já existe.*

Alguns pacifistas insurgem-se contra a intenção (patética) de fabricarmos bombas atômicas no Brasil, manifestada recentemente por algumas autoridades. Muito, mas muito mais ameaçador do que isto é o projeto de centralização, em um único banco de dados, de **todas** *as informações referentes à ficha individual de* **todos** *os brasileiros.*

Se este projeto for levado adiante, a maior rapidez na investigação de alguns eventuais criminosos implicará no **controle total e absoluto** *sobre* **todos os cidadãos.** *Nestas condições o computador se tornará uma arma muito mais letal que a própria bomba atômica. Será dado mais um passo no sentido de alienar o indivíduo do poder de decisão sobre sua própria vida e sobre o futuro de seus descendentes.*

Se isto acontecer, poderemos dizer que o advento do computador realmente propiciou a chegada de 1984.

Feliz Ano Novo!

CARTAS DOS LEITORES

Caros amigos,

. . . O que realmente me levou a escrever para vocês foi o artigo **LGM — Mensagem de Vega** — Quebra-Cabeça da revista número cinco. No rodapé da página 22 existe uma nota fazendo menção à uma organização chamada **Mensa International** que muito me chamou a atenção. Como eu poderia obter maiores informações a respeito dela? Na ocasião vocês afirmaram que, escrevendo para esta seção seria possível obter dados a respeito desta organização. Gostaria muito que me enviassem estas informações, assim que possível.

Desde já agradeço a sua atenção, certo que serei atendido. **Fabio Appolinario.** São Paulo, Capital.

Prezado Fabio,

Obrigado pelos parabéns. A qualidade da **Microhobby** *é nossa maior preocupação e é através das cartas de nossos leitores que sabemos se estamos no caminho certo ou não.*

Para evitarmos responder a todos os leitores, individualmente, decidimos publicar brevemente uma matéria sobre a **Mensa.** *Pedimo-lhe um pouco de paciência para que possamos elaborá-la.*

Prezados Senhores,

Nós, sofredores do TK, ao passarmos um programa para o gravador ou vice-versa, estamos ávidos para que haja uma solução para este sofrimento.

Por várias vezes, a revista anunciou "a solução", como por exemplo, a utilização do Tig-Loader (página inteira, a cores na número cinco).

Bem, comprei o tal aparelho, porém não obtive resultado positivo algum, a não ser que fiquei 15 mil cruzeiros mais pobre.

Agora, a revista anuncia outro "salvador", o CMS/ZX, por um preço camarada de apenas Cr$ 19.950,00. Pergunto: "será que este realmente vai funcionar? Qual a sua súgestão para que eu não corra outro risco?"

Eu sugiro que, para esta revista manter sua boa reputação, publique somente este tipo de anúncio depois dos aparelhos devidamente testados ou então colocar em baixo do anúncio: "Não testado pela revista".

Considero a revista **Microhobby** excelente e gostaria de manter esta impressão.

Quero deixar claro que o TK é um computador com aplicações infinitas e estou satisfeito com o mesmo. **Willem H.J.G. Scheepmaker,** São Paulo, Capital.

Caro Willem,

Diariamente você ouve e vê no rádio, na televisão, nos jornais e revistas, nos out-door's e luminosos, anúncios afirmando por exemplo, que é muito bom fumar cigarros, beber refrigerantes, transmitindo a idéia de que estes hábitos tornam as pessoas mais jovens, inteligentes e bonitas. Porém, nem por isso estes meios de comunicação são responsáveis pela veracidade dos anúncios pagos.

CARTA DO EDITOR

Prezado Leitor:

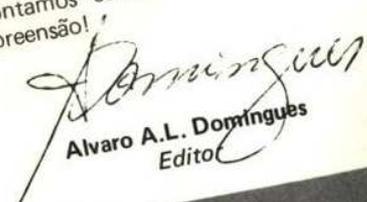
Você deve ter notado algumas modificações (para melhor, espero) neste número da *Microhobby*. Estas modificações visam oferecer-lhe uma revista melhor do que a que você tem recebido. Não paremos por aí. Certamente, para breve, muitas coisas novas serão incorporadas na sua revista.

Também estão sendo feitas algumas alterações para melhor atendê-lo, tanto como leitor como eventual colaborador.

Por outro lado, gostaríamos que você ajudasse um pouco, tomando algumas atitudes que facilitariam enormemente nosso trabalho:

a) **Se você for nos escrever, não misture assuntos em sua carta.** É comum recebermos cartas de leitores onde aparecem dúvidas para serem respondidas pela seção **Desgrilando,** uma colaboração, críticas e sugestões, uma reclamação sobre assinatura ou brinde, tudo na mesma carta. Se você por acaso tiver mais de um assunto, separe em três cartas no mesmo envelope: uma para assuntos relativos à assinatura (com exceção do pedido), uma para dúvidas, sugestões e críticas e uma para colaborações. Pedidos de assinatura, compra de livros e números atrasados deverão ser encaminhados em outro envelope. Assuntos relativos a anúncios e anunciantes (com exceção dos pequenos anúncios), deverão estar em outro envelope. Pequenos anúncios deverão ser enviados também em um envelope separado. Identifique os envelopes: Redação, Assinaturas, Publicidade e Pequenos anúncios.

b) **Evite usar nosso telefone.** Embora possamos *eventualmente* atender suas dúvidas por telefone, não é este nosso propósito. Temos que fazer a *sua* revista e não dispomos de tempo suficiente para responder a todos os telefonemas que recebemos. Assim, reserve seu telefone para dúvidas sobre *textos* e *programas* já publicados na *Microhobby*. As dúvidas devem ser específicas e objetivas. ***Não nos ligue*** sem tentar, você mesmo, resolver o problema. As reclamações, quanto ao recebimento da revista e/ou brinde, serão encaminhadas ao departamento responsável.

Contamos com a sua colaboração e compreensão!

Alvaro A.L. Domingues
Editor

Não estamos dizendo com isto que os anúncios existentes na **Microhobby** *não são confiáveis, apenas queremos deixar claro que a responsabilidade pela qualidade dos produtos anunciados é do* **anunciante!!**

Especificamente com relação ao **Tig-Loader**, *informamo-lhe que em muitos casos, ele é a nossa "salvação" quando tentamos introduzir programas de fitas enviadas pelos leitores. Claro que há casos em que só mesmo com um "milagre" (coisa que os aparelhos não sabem fazer, ainda!) conseguimos introduzí-los.*

Infelizmente, não possuimos um CMS/ZX e por esta razão não o testamos, mas acreditamos não haver problemas com seu funcionamento.

Os Oitenta e Por Dentro do Apple

Prezados Senhores,

Como usuário de TK, fiquei muito satisfeito com o lançamento de uma revista destinada aos usuários destas máquinas; não tive dúvidas, tratei logo de oficializar minha assinatura, pensando no conteúdo das seções, bem como nas seções previstas.

Gostei muito dos números iniciais, mas gostaria de transmitir a minha insatisfação e creio também que deva ser a de grande maioria dos assinantes.

1. Na seção **Hobby** ainda não foi publicado nenhum projeto, como se trata de uma revista que conta com a colaboração dos usuários para beneficiar usuários, não deveria haver problemas na montagem desta seção; afinal não se trata de divulgar segredo industrial.

2. Assuntos prometidos para o número seguinte e que não são publicados; exemplo o artigo "Funções especiais para o TK-85", na revista número 4.

3. Assuntos que nada têm a ver com o TK, no caso "Por Dentro do Apple"; trata-se de tirar esta seção, uma vez que o assunto não é próprio; a revista **Microhobby** é para os usuários do TK. Sugestão: pensem numa revista para usuários desta máquina.

Por fim, gostaria que os senhores encarassem isto como uma crítica construtiva visto que, o que foi proposto não está sendo cumprido. **Rubens Maus** — Curitiba, PR.

Caro Rubens,

Provavelmente sua preocupação com as seções sobre os micros das linhas Apple e TRS-80 é comum a muitos leitores (inclusive nós mesmos que, em nossos lares possuímos também, micros TK). Nossa idéia parece não ter sido entendida completamente por você. (Vide a seção **Desgrilando**). *Também é comum recebermos cartas de leitores que têm acesso a outros micros pedindo-nos informações sobre o funcionamento dos mesmos. O número destas cartas não é desprezível. Justifica-se, portanto, a introdução das duas (e agora três) seções, uma vez que atende às necessidades de* **usuários do TK.** *Além disso, mesmo para aqueles que só têm acesso ao TK, cremos ser importante (seja como curiosidade ou como aperfeiçoamento técnico) conhecer o funcionamento de outras máquinas — é importante não "bitolar-se"!*

Imagine que você veja, em algum lugar, um excelente programa para o TRS-80 ou para o Apple e o queira utilizar no TK. Ao invés de escrever-nos, perguntando como proceder, acompanhe as seções sobre esses dois micros e você mesmo terá condições de adaptá-lo! Para facilitar sua tarefa, introduziremos a seção **Vice-Versa** *nos próximos números. Nessa seção serão expostas formas para traduzir-se programas do Basic-TK para os do TRS-80, Apple e vice-versa.*

Não estaremos tomando espaço do TK, uma vez que a quantidade de matérias por edição está cada vez maior (estamos introduzindo três colunas em algumas matérias). Com relação a seção **Hobby**, *como você pode notar, começamos a* **tirar o atraso** *(pelo qual nos desculpamos) neste número.*

Quanto aos assuntos prometidos, só podemos dizer que eles foram efetivamente publicados, ainda que com um número de atraso. Nós **prometemos** *tomar cuidado para que não ocorram novos enganos.*

Suas críticas foram realmente construtivas e esperamos recebê-las, sempre que necessário, de nossos leitores. Estamos imensamente agradecidos.

O TK e as antenas parabólicas

Prezados Senhores,

Fiquei muito interessado com o Quebra-Cabeça da **Microhobby** número cinco. Tenho 15 anos, gosto muito de eletrônica, principalmente de telecomunicações. Antes de comprar o meu TK-82C, tinha um par de walkies-talkies e resolvi fazer uma antena para eles. Com a antena que fiz, eles ficaram muito potentes em recebimento de ondas — eu escutava estações de todo o mundo. Num sábado à tarde, liguei meu rádio e recebi sinais muito estranhos e os gravei. Não sei o que podem ser.

Agora comprei um TK e gostaria de saber como fazer para ligar uma antena nele. . . **João Batista Donegá Filho,** Mogi Guaçu, SP.

Caro João Batista,

Tínhamos a intenção de colocá-lo em contato com Nabor Rosenthal, mas infelizmente ele viajou para a Índia e deverá retornar somente o ano que vem.

Com relação a MENSA, pedimo-lhe que leia a resposta à carta do **Fabio Appolinário.**

Reversão do vídeo no CP-200 e Joystick

Prezados Senhores,

Em primeiro lugar, gostaria de parabenizá-los pelo ótimo conteúdo da revista **Microhobby** até o presente momento, principalmente pelas dicas e programas apresentados, que são úteis e de aplicação imediata, como no caso do programa **PentaSpeed** (publicado na revista número 4) e que efetivamente traz um bom ganho de tempo na gravação e leitura de programas.

Em segundo lugar, gostaria de agradecer o envio da fita brinde. Um alerta apenas com relação a qualidade da fita utilizada, pois somente após três dias de muitas tentativas, consegui carregar corretamente o programa **Mansão Maluca,** devido ao fato da fita trancar no cassete e prejudicar a rotação da mesma no gravador.

Finalizando, apreciaria receber, caso possível, duas dicas a respeito de duas dúvidas:

A primeira é com relação a uma fita **TKDREZ I** da Microsoft, adquirida recentemente. Ocorre que, ao rodar a mesmo num micro CP-200 da Prológica, obtenho a imagem do tabuleiro de xadrez invertida na tela. Fui informado que isto ocorre pelo fato do jogo ter sido desenvolvido pela Microsoft para o TK-82C, que trabalha com inversão de vídeo. Desejo saber se é possível efetuar uma inversão de vídeo através de **soft** na fita **TKDREZ I**, sem danificar a mesma, ou se é necessário alguma adaptação no micro.

A segunda é com relação ao uso de joystick. Desejo saber, se no CP-200 é possível a utilização de joystick, ou se é necessário, também, fazer alguma adaptação no micro. **Luiz Fernando Scheffel** — Porto Alegre, RS.

Caro Luis Fernando,

Ficamos gratos por seus elogios.

Com relação às suas dúvidas, infelizmente a solução menos desejável é a única que existe. A reversão do tabuleiro só pode ser feita com adaptações no CP-200. Se você quiser, procure na seção **pequenos anúncios** *as informações sobre quem pode fazer essa adaptação para você. Quanto ao joystick, a Prológica desenvolve um especialmente desenhado para o CP-200. Procure no revendedor que lhe vendeu o CP-200, provavelmente ele terá este módulo.*

Arquivo

Prezados Senhores,

. . . o número 3 de **Microhobby** está, como já me acostumei a constatar, melhor que o anterior. Parabéns!

Parece que o programa **Apagando a Tela** (revista número 3) veio a responder minha perguntas sobre as linhas REM. Em compensação tenho novamente dúvidas, ambas referentes ao programa **Arquivo** (revista número 3):

1) Por que as linhas INPUT A são seguidas de LET A = INT A?

2) Por que o programa pede para digitar qualquer tecla (linha 489Ø) se após a pausa segue um RETURN incondicional? E ainda, de novo PAUSE sem POKE?

Gostei de ver o programa de matemática financeira (página 10), uma idéia que já me ocorrera, mas ainda não conseguira desenvolver. Como o Flavio Rossini colocou os pontos e as vírgulas nos seus devidos lugares, o que era a parte mais difícil, mas deixou os "milhões" na mesma coluna dos "milhares", vai aqui minha modesta colaboração:

O PRINT de saída deveria ser feito não em função de "L" mas de "U", que é a variável que "conta" os algoritmos da parte inteira de "N", assim a linha 121Ø seria modificada para:

121Ø PRINT "CR $"; TAB (21-U); N$

E para contar também os espaços ocupados pelos pontos deveríamos acrescentar a linha:

1178 LET U = U + 1

Obviamente a inscrição "CR $" pode ser omitida ou substituída por outra. E na "TAB (21-U)" pode ser usado um número menor se quisermos "escrever" mais à esquerda.

Por enquanto é só! . . . **Alejandro Luis Cobos** — São Paulo, Capital.

Caro Alejandro,

É com imenso prazer que tornamos a responder uma carta sua. As perguntas que você nos faz são respondidas abaixo:

1) A linha LET A = INT A seguindo INPUT A é apenas para evitar que o programa pare se o usuário distrair-se e digitar algum número com digitos não nulos após a vírgula.

2) A instrução PAUSE 3ØØØØ produz uma **pausa** *de pouco menos de nove minutos. Quando digita-se qualquer tecla,*

menos BREAK, a pausa é interrompida e o programa continua a rodar. Experimente substituir as linhas:

```
4900 PAUSE 30000
```

por:

```
4900 IF INKEY$="" THEN GOTO 4900
```

Com relação ao fato de **Pause** *não estar acompanhado de POKE 16497,255 temos a dizer que não há nenhum problema. Apenas os primeiros micros do TK necessitam disso.*

Aproveitamos, também, para agradecer as sugestões ao programa do artigo **Matemática Financeira em Basic** *de Flavio Rossini.*

Hexamem

Prezados Senhores,

Possuo um TK-85 de 48K e venho acompanhando com grande interesse as aulas de Assembly na revista **Microhobby,** sendo que, na número 4 (aula número 3) deparei-me com um "grilo" na execução do programa **Hexamem** do qual tirei uma listagem e cópias da tela para que V.Sas. me expliquem o que ocorreu.

Primeiramente, para rodar o seu programa tive que "enxertar" a linha 1Ø2 SCROLL, pois antes surgia a indicação da tela cheia.

Como V.Sas. poderão observar, o programa foi rodado com o exemplo da lição e ao fazer o comando RUN 3ØØØ, o resultado foi diferente do que está indicado e era esperado, por exemplo, a memória 3ØØØØ que deveria registrar 164 e registrou 68.

Parece-me que não houve erro na digitação do programa, e isso me deixou "grilado". Gostaria de que verificasse o que ocorreu, escrevendo-me ou explicando o fato na próxima lição.

Gostaria de saber também o que faz as linhas 8Ø e 85. Fiz um teste dando entrada em A$ = "XS" e meu programa saiu do ar.

Na aula número 2 (revista número 3) fêz-se referência a um livro (". . . 16 K para que se possa executar os programas mais interessantes deste livro. . ."). Gostaria de saber se vocês já o editaram e, caso positivo, como posso adquirí-lo.

Agradeço-lhe antecipadamente, e nesta oportunidade valho-me do ensejo para parabenizá-los pelo excelente nível de ensino e pela excelente didática de suas aulas publicadas. **Joaquim J. Mariano,** Goiânia, GO.

Caro Joaquim,

Cremos que você teve que introduzir a linha 1Ø2 SCROLL por **erro de digitação,** *pois na linha 6Ø já existe o comando SCROLL. O mesmo deve ter ocorrido quando você introduziu os digitos A4 (hexadecimal). O código 68 que você obteve na memória 3ØØØØ corresponde a 44 (em hexadecimal) e é provável que você tenha digitado 44 ao invés de A4.*

*As linhas 8Ø e 85 (e também 13Ø, 135, 14Ø e 145) tem sua utilidade esclarecida na lição número 4 do curso de Assembly (***Microhobby** *número 5). De qualquer forma, seu programa não deveria ter "saído do ar" como aconteceu. Novamente, você deve ter digitado algo errado numa das linhas citadas acima. Se nós estivermos enganados, então seu TK-85 está com defeito.*

O livro referido é o **Linguagem de Máquina para o TK — volume I"** *que já foi editado. Você pode adquirí-lo enviando um cheque nominal e cruzado ou um vale postal no valor de Cr$ 8.000,00 para* **Micromega P.M.D. Ltda. Cx. Postal 54121, CEP 05096** *— São Paulo, SP. Junto ao cheque, anexe seu nome, endereço completo e o motivo (livro) pelo qual você o está enviando.* →

Traduções para o TK

Prezados Senhores,

... gostaria que me orientassem na transcrição dos comandos (funções sublinhadas na xerox anexa) para o Basic do TK-85.

... 20 T = O

60 DIM V(3), X(3), P(3), R(3,3)

70 MAT V = ZER

80 MAT X = ZER

90 MAT P = ZER

100 MAT R = ZER

110 DATA 1,2,2,3,3,1,1,3,3,2,2,1,2,3, 3,1,1,2,0

120 PRINT "WELLCOME TO 'WAR 3'. TWO OR THREE HUMANS MAY PLAY!"

130 PRINT "DO YOU WISH SOME ASSISTANCE";...

... 290 FOR J = 1 TO N1

300 READ A,B

310 PRINT "DISTANCE" (FT.) "; A;" TO "; B;

320 INPUT R (A,B)

330 R(B,A) = R(A,B)

340 NEXT J

350 PRINT " "

360 RESTORE

370 IF N = 2 THEN 460

380 FOR J = 1 TO N

390 READ A,B,C,D,E,F

400 IF R (A,B) < R(C,D)+R(E,F) THEN 440

410 PRINT "ERROR — ILLEGAL TRIANGLE. RE-ENTER RANGES."

420 RESTORE

430 GOTO 290

440 NEXT J ...

Grato e parabéns. **Paulo Roberto B. Bergo,** São José dos Campos, SP.

Caro Paulo Roberto,

Ficamos muito gratos pelos seus elogios, pois é motivo de orgulho para nós, sabermos que estamos agradando.

Quando as informações sobre as TPlak's, você as encontrará nos números anteriores.

As linhas que você sublinhou em seu programa são de fácil adaptação para o Basic-TK e se você conseguir traduzir completamente o programa, gostaríamos que o enviasse para ser publicado em nossa revista.

Você sublinhou as linhas:

70 MAT V = ZER

80 MAT X = ZER

90 MAT P = ZER

110 DATA 1,2,2,3,3,1,1,3,3,2,2,1,2,3, 3,1,1,2,0

Além dessas linhas, teremos que traduzir também as que seguem:

60 DIM V(3), X(3), P(3), R(3,3)

100 MAT R = ZER

300 READ A,B

360 RESTORE

390 READ A,B,C,D,E,F

420 RESTORE

A adaptação para o TK, fica assim:

```
 60 DIM V(3)
 61 DIM X(3)
 62 DIM P(3)
 63 DIM R(3,3)
 64 FOR Z=1 TO 3
 65 LET V(Z)=0
 66 LET X(Z)=V(Z)
 67 LET P(Z)=V(Z)
 70 FOR Y=1 TO 3
 80 LET R(Z,Y)=V(Z)
 90 NEXT Y
100 NEXT Z
110 LET D$="1223311332212331120
"
285 LET Z=1
300 LET A=VAL D$(Z)
303 LET B=VAL D$(Z+1)
306 LET Z=Z+1
350 LET Z=1
390 LET A=VAL D$(Z)
391 LET B=VAL D$(Z+1)
392 LET C=VAL D$(Z+2)
393 LET D=VAL D$(Z+3)
394 LET E=VAL D$(Z+4)
395 LET F=VAL D$(Z+5)
395 LET Z=Z+6
420 LET Z=1
```

Esperamos que você tenha êxito em seu empreendimento e aguardamos ansiosamente os resultados. ○

DICAS

A LINHA REM

Nancy Mitie Ariga

Muitas vezes vamos digitar um programa em linguagem de máquina e se torna necessário criar inicialmente, uma linha **REM** com muitos pontos, o que é um tanto quanto trabalhoso.

A *dica* deste mês é um programa capaz de criar esta linha **REM**, evitando assim, que você fique digitando "n" pontos. . .

Inicialmente digite o programa 1 (à linha **REM** deve ter 96 pontos).

```
   1 REM ......................
..................................
..................................
........
   2 FOR I=16514 TO 16609
   3 SCROLL
   4 PRINT I;"..";
   5 INPUT X
   6 PRINT X
   7 POKE I,X
   8 NEXT I
```

PROGRAMA 1

Aperte a tecla **RUN** e digite os números da listagem decimal a seguir:

```
16514..0        16515..0
16516..205      16517..35
16518..15       16519..1
16520..6        16521..0
16522..42       16523..130
16524..64       16525..229
16526..9        16527..68
16528..77       16529..42
16530..41       16531..64
16532..9        16533..34
16534..41       16535..64
16536..33       16537..12
16538..64       16539..62
16540..9        16541..94
16542..35       16543..86
16544..213      16545..235
16546..9        16547..235
16548..114      16549..43
16550..115      16551..35
16552..35       16553..61
16554..40       16555..3
16556..209      16557..24
16558..238      16559..225
16560..229      16561..1
16562..226      16563..64
16564..167      16565..237
16566..66       16567..68
16568..77       16569..225
16570..237      16571..184
16572..33       16573..227
16574..64       16575..54
16576..0        16577..35
16578..54       16579..2
16580..35       16581..193
16582..3        16583..3
16584..113      16585..35
16586..112      16587..35
16588..54       16589..234
16590..11       16591..11
16592..35       16593..17
16594..1        16595..0
16596..235      16597..25
16598..235      16599..54
16600..27       16601..237
16602..175      16603..54
16604..117      16605..52
16606..205      16607..43
16608..15       16609..201
```

LISTAGEM DECIMAL

Ao terminar de digitar a listagem, peça ao computador para mostrar a listagem, por meio de tecla **LIST**. A linha 1 **REM** contém vários caracteres que representam o programa em linguagem de máquina, que criará a sua linha **REM**.

```
   1 REM   LN ??" " E RND FAST ■?
?EDRND■6DRND5£RNDY■???STR$ FOR ■
 FOR ?F??7XC■SGN / INPUT LPRINT
FAST " LLIST RND■ GOSUB PI?? LPR
INT GOSUB ■5 STOP RND0 70 "7AT ■
???70 REM ""7)"  FOR ; FOR 0. GO
SUB ■0?OLN F?TAN
   2 FOR I=16514 TO 16609
   3 SCROLL
   4 PRINT I;"..";
   5 INPUT X
   6 PRINT X
   7 POKE I,X
   8 NEXT I
```

Retire as linhas 2 a 8 e digite as novas linhas do programa 2.

```
   1 REM   LN ??" " E RND FAST ■?
?EDRND■6DRND5£RNDY■???STR$ FOR ■
 FOR ?F??7XC■SGN / INPUT LPRINT
FAST " LLIST RND■ GOSUB PI?? LPR
INT GOSUB ■5 STOP RND0 70 "7AT ■
???70 REM ""7)"  FOR ; FOR 0. GO
SUB ■0?OLN F?TAN
   3 PRINT "COMPRIMENTO DA REM:"

   4 INPUT C
   5 PRINT C
   6 PAUSE 60
   7 POKE 16514,C-256*INT (C/256

   8 POKE 16515,INT (C/256)
   9 RAND USR 16516
  10 CLS
  11 LIST 2
```

PROGRAMA 2

Vamos ver como este programa funciona?

Ponha o programa para funcionar. A seguir, digite o comprimento que a sua linha **REM** deveria ter e, num piscar de olhos, o programa gera uma linha 2 **REM**, com o número de pontos que você desejava.

```
   1 REM = LN ??" " E RND FAST ■?
?EDRND■6DRND5£RNDY■???STR$ FOR ■
 FOR ?F??7XC■SGN / INPUT LPRINT
FAST " LLIST RND■ GOSUB PI?? LPR
INT GOSUB ■5 STOP RND0 70 "7AT ■
???70 REM ""7)"  FOR ; FOR 0. GO
SUB ■0?OLN F?TAN
   2 REM .......................
   3 PRINT "COMPRIMENTO DA REM:"

   4 INPUT C
   5 PRINT C
   6 PAUSE 60
   7 POKE 16514,C-256*INT (C/356

   8 POKE 16515,INT (C/296)
   9 RAND USR 16516
  10 CLS
  11 LIST 2
```

Agora que você já testou o programa 2 e sabe como ele funciona, retire a linha 2 e grave o programa em uma fita de boa qualidade.

E toda vez que você for digitar um programa em linguagem de máquina muito longo, carregue este programa e digite o número de pontos que precisar. Depois, é só retirar as linhas desnecessárias ao programa que você vai digitar. ○

PEQUENOS ANÚNCIOS

— TROCO **Programas** nacionais e importados (recuso qualquer pagamento em dinheiro) para o TK-82, 83 e 85. Tratar com Elington pela Caixa Postal 39003 — CEP 21490.

— VENDO diversos solos em linguagem de máquina, para o TK-82C e TK-85. Tratar com Marco Antonio pelo tel.: 225-7507 — Belo Horizonte — MG.

— TROCO **Programas** (jogos) e idéias para o TK. Tratar com José Augusto S. de Campos. Av. Dr. João S. Alves de Carvalho, nº 530 — Jardim Ipiranga — Porto Alegre — RS — CEP: 90000. Mandar dados pessoais e programas.

— VENDO super TK com **high speed,** gravação independente de dados, inversão de vídeo por soft ou chave, controle remoto do gravador (soft ou chave), tecla reset alta-resolução, 16K e função slow, junto com programas em linguagem de máquina. Jogos e aplicativos. Informações com César pelo telefone: 201-8409 (nos fins de semana).

— PROCURO LEITOR que tenha esquema da interface e som para o TK, ou alguém que o construa. Esquemas e/ou propostas para Paulo A. Fessel. R. General Góis Monteiro, 113 — Piracicaba — SP. Também procuro pessoas que tenham o nº 1 da Microhobby, para enviar-me duas cópias xerox no mesmo endereço.

— VENDO TK-85 com alta resolução jogos, joystick, tudo por Cr$ 250.000,00. Três meses de uso. Tratar com Edson. CEP: 04437. R. Manoel Vaz, 265 — Jd. Consórcio — Sto. Amaro, SP.

- VENDO ou TROCO programas para os similares de CP 200, tais como: TK-82, 83, 85, e outros , aos interessados escrever para: Adelson de Carvalho A. Jr. Rua da Hora, 465 — apto. 102-B — Recife, PE — CEP: 50000.

— Making The Most of Your ZX-81. Getting Acquainted with your ZX-81. Tratar com Edison — Fone: 235-3479. Hor. Comercial — Cx. Postal 7252 — CEP: 01031 — SP.

— COMPRO Interface para máquina elétrica (datilografia-impressora). Propostas para Roberto de A. Santos. Rua Uruguai, 205, apto. 803 — Tijuca, RJ — CEP: 20510.

— Estou interessado em comprar uma TK-Printer usada. Comunicar-se com Rubens Maus — Rua Luiz Dallagassa, 208 Vila Haver — CEP: 80000 — Curitiba, PR.

— TROCO **Programas** para Micros TK-82C e similares. Possuo o Frogger, Mazógs, Snake Bite, Galaxy e outros. Tratar pelo tel.: 853-4717 (c/ Sergio) — São Paulo — SP.

— Desejo corresponder-me com usuários de micros similares ao TK82-C residentes em Blumenau, para TROCA de programas. Escrever para Zoltan Bergman — Cx. Postal 2172 — CEP 89100 — BLUMENAU, SC.

CURSO DE BASIC TK
aula 7

Pierluigi Piazzi
Flavio Rossini

VARIÁVEIS INDEXADAS MATRIZES – FLAGS

Na aula número 1 introduzimos a palavra-chave LET, que associava à uma variável numérica um "lugar" na memória. Entretanto, existem ocasiões em que é desejável armazenar vários valores para uma "mesma" variável, distinguindo seus diferentes valores através de ***índices*** entre parênteses; seria como se construíssemos uma ***tabela*** na memória, à qual denominamos ***matriz***. Assim, temos a palavra-chave DIM (tecla D) cuja função é reservar espaço na memória para os valores que desejamos colocar, atribuindo inicialmente valor ***zero*** a todos eles. Experimente então o seguinte programa:

```
 10 SLOW
 20 DIM X(5)
 30 FOR I=1 TO 5
 40 PRINT "X(";I;")=";X(I)
 50 NEXT I
 60 PRINT
 70 FOR I=1 TO 5
 80 PRINT "X(";I;")=?"
 90 INPUT X(I)
100 PRINT AT I+5,5;X(I)
110 NEXT I
```

Na instrução 2Ø reservamos espaço na memória para cinco valores de ***X*** colocando inicialmente valor Ø (zero) para os mesmos; a seguir, usando INPUT, associamos valores aos vários ***X***'s e usando um "pequeno truque" na linha 1ØØ escrevemos, na tela, os números colocados.

Note que há diferença de significado dos índices: com a palavra-chave DIM ele indica o ***tamanho máximo*** da tabela; nas demais instruções, ele indica a ***posição*** na tabela.

O que acontece se, no programa anterior, colocamos a linha:

```
95 DIM X(5)
```

?

Infelizmente no Basic-TK as variáveis indexadas têm uma limitação: ao contrário das variáveis normais, seu "nome" pode ter apenas uma letra; assim, por exemplo, o seguinte comando é inválido:

```
DIM PAZ(34)
```

Em contrapartida, é possível fazer ***matrizes*** (TABELAS) com mais de uma dimensão (até 255 dimensões). Por exemplo, duas dimensões são facilmente imagináveis se pensarmos em linhas e colunas.

Por exemplo:

```
 10 DIM A(3,4)
 20 FOR I=1 TO 3
 30 FOR J=1 TO 4
 40 PRINT AT 4,4;"A(";I;",";J;"
=?"
 50 INPUT A(I,J)
 60 PRINT AT 4,11;A(I,J)
 70 PAUSE 90
 80 CLS
 90 NEXT J
100 NEXT I
110 PRINT TAB 4;"MATRIZ ESCOLHI
DA"
120 PRINT
130 FOR I=1 TO 3
140 FOR J=1 TO 4
150 PRINT AT 2*I+2,4*J;A(I,J)
160 NEXT J
170 NEXT I
```

Repare que na linha 1Ø reservamos espaço para uma tabela de três linhas e quatro colunas, ou seja, para 3x4 = 12 variáveis com valor inicial zero.

A seguir, você deve fornecer os valores para cada variável (INPUT) e finalmente a ***matriz*** é colocada na tela.

Note que, estamos usando um ***loop*** dentro de outro ***loop:*** desta maneira, o ***loop*** interno será repetido tantas vezes quantas forem indicadas pelo ***loop*** externo. Note que os NEXT's devem aparecer em ordem inversa aos FORs, ou seja:

```
┌FOR I = . . .
│┌FOR J = . . .
│└NEXT J
└NEXT I
```

Repare no uso da instrução AT para a formatação da saída na linha 15Ø; com esta formatação, qual o número máximo de algarismos que posso usar para cada valor? É possível aumentá-lo? Você saberia "imaginar" uma matriz com três ou mais dimensões?

Noção de FLAG: utilização na ordenação de números

Denominamos FLAG a uma ***variável auxiliar*** cuja função é indicar a ocorrência ou não de determinada condição durante a execução de um programa. Uma utilização prática para este tipo de variável, está na ajuda em descobrir "erros" em programas razoavelmente complicados, principalmente aqueles com um grande número de instruções IF. Nestes programas, a quantidade de "caminhos" possíveis é muito grande, o que dificulta a análise do mesmo, "passo a passo", conforme fluxograma ao lado.

Que corresponde a:

```
 10 SLOW
 20 PRINT AT 4,3;"QUANTOS NUMER
OS DEVO ORDENAR?MAX.(22)"
 30 INPUT N
 40 DIM N(N)
 45 CLS
 50 FOR I=1 TO N
 60 PRINT AT 4,4;"N(";I;")=?"
 70 INPUT N(I)
 80 NEXT I
 90 FAST
100 CLS
105 LET F=0
110 FOR I=1 TO N-1
120 IF N(I)<=N(I+1) THEN GOTO 1
70
130 LET F=1
140 LET B=N(I)
150 LET N(I)=N(I+1)
160 LET N(I+1)=B
170 NEXT I
180 IF F=1 THEN GOTO 105
190 SLOW
200 FOR I=1 TO N
210 PRINT TAB 3;N(I)
220 NEXT I
```

Repare o uso do DIM (linha 2Ø) para reservar espaço na memória para 22 números que deverão ser fornecidos ao computador através do INPUT na linha 7Ø. Note a mudança de velocidades de SLOW para FAST e, a seguir para SLOW;

isto porque a tela não é necessária enquanto o computador faz a "ordenação". Note também que o ***loop*** de ordenação faz o contador I variar de ***1*** até ***N—1***, pois caso contrário, ao chegar no último número, iríamos obter um erro (Por que?).

Finalmente, repare no uso da variável *B* como auxiliar para fazer as trocas e o uso do mesmo nome (N) para a variável indexada, e para a variável que indica quantos números devem ser ordenados; elas são distinguidas pelo computador, pois esta última não possui índices entre parênteses.

Para finalizar a aula, apenas uma observação quanto aos índices: eles devem ser sempre números inteiros e positivos.

Exercícios:

1. Altere o último programa da aula para que imprima uma mensagem de erro se na linha 3Ø for introduzido um número maior que 22. Além disso, faça com que ele coloque os números em ordem decrescente.

2. Elabore um programa que, dada uma matriz bidimensional quadrada (nº de linhas = nº de colunas) calcule seu determinante.

3. Faça um programa que seja capaz de "ler" duas matrizes de duas dimensões e calcule a soma e diferença "imprimindo" na tela a matriz-resultado. O

FLUXOGRAMA

RESPOSTAS DOS EXERCÍCIOS DO CURSO DE BASIC

A partir deste número de **Microhobby** *passaremos a publicar as respostas do Curso de Basic continuamente. Nesta edição, vocês terão as respostas dos exercícios desde o início, ou seja, a partir da revista número dois até a última edição, número 6.*

As respostas dos exercícios desta edição, publicaremos na próxima revista (oito) de **Microhobby.** *Assim sendo, nossos leitores ficarão sempre em dia com o Basic.*

Aula 2

1)

```
10 LET HOPE=6
20 PRINT "HOPE = ";HOPE
30 PRINT "HOPE + 2 = ";HOPE+2
40 PRINT "3*HOPE = ";3*HOPE
50 PRINT "5+7 = ",5+7
60 GOTO 50
```

2) a — Ele imprime o valor de **A** que é igual a **5,** e o valor de **B** que é igual a 1Ø.

Apesar do valor de **B** não estar no programa, ele está na memória do computador.

b — O **micro** acusará o erro na linha 2ØØ, pois a linha 15Ø "limpou" a memória do micro.

c — Se for acrescentada a linha 4ØØ NEW, o micro imprimirá os valores de **A** a **B** e **limpará** o programa da memória.

Aula 3

1)

```
10 FOR A=0 TO 21
20 PRINT TAB A;A
30 NEXT A
```

2)

```
10 FOR A=1 TO 21
20 PRINT TAB A;"[illegible]"
30 NEXT A
```

3) a —

```
10 RAND
20 PRINT "LOTERIA"
30 FOR A=1 TO 13
40 PRINT INT (RND*3)+1
50 NEXT A
```

b —

```
10 RAND
20 PRINT "LOTERIA"
30 LET A=1
40 FOR B=1 TO 13
50 LET C=INT (RND*3)+1
60 PRINT "JOGO ";A,"COLUNA ";C
70 LET A=A+1
80 NEXT B
```

c —

```
10 RAND
20 PRINT "LOTERIA";TAB 10;1;TA
B 15;"X";TAB 20;2
30 FOR A=1 TO 13
40 LET T=10+5*(INT (RND*3))
50 PRINT "JOGO ";A;TAB T;"X"
60 NEXT A
```

4)

```
10 LET I=10
20 LET F=10
30 PRINT AT I,F;"*"
40 LET F=F+1
45 IF F<32 THEN GOTO 30
50 LET I=I+1
55 LET F=0
60 GOTO 30
```

Aula 6

1)

```
10 INPUT N
20 IF N<0 OR N<>INT N THEN GOT
O 80
30 LET M=1
35 LET I=N
40 LET M=M*I
50 LET I=I-1
60 IF I=1 THEN PRINT "FAT. ";N
;"=";M
70 IF I=1 THEN STOP
75 GOTO 40
80 PRINT "NUMERO NAO ACEITO"
90 GOTO 10
```

2) a —

```
10 INPUT N
20 IF N-2*INT (N/2)=0 THEN PRI
NT N;" E* PAR"
30 IF N-2*INT (N/2)=1 THEN PRI
NT N;" E* IMPAR"
```

b —

```
10 INPUT N
20 IF N-6*INT (N/6)=0 THEN PRI
NT N;"  E* MULTIPLO DE 6"
30 IF N-6*INT (N/6)<>0 THEN PR
INT N;" NAO E* MULTIPLO DE 6"
```

O

MINICALC

Nancy Mitie Ariga

Atualmente no mercado estão à venda uma grande variedade de Visicalcs, que são planilhas eletrônicas que possibilitam a manipulação de informações com maior facilidade e ocupando o menor espaço na memória do computador.

Apresentamos aqui um Minicalc, que é uma redução dos Visicalcs que estão à venda. Este Minicalc foi planejado para o controle de um orçamento doméstico (calcula o total das despesas e o saldo do mês).

Para iniciar o programa (veja programa na página seguinte), digite **RUN** e aparecerá na tela uma parte da planilha. Importante: na tela aparecerão sempre as colunas das Contas de dois meses.

```
CONTAS        JAN        FEV
```

```
ENTRADA DE DADOS 0,1
"L"
```

A planilha está dividida em várias células, que estão organizadas por coordenadas (coluna, linha). Por exemplo: a conta ENTRADA corresponde ao dado (0,1); a conta LUZ (0,4), o mês de JANEIRO corresponde à coluna 1, e assim por diante.

Primeiramente, digite as contas nas células da coluna "Ø". Sua planilha ficará assim:

```
  CONTAS        JAN        FEV
1 ENTRADA
2 SALD.ANT.
3 TOT.ENTR.
4 LUZ
5 AGUA
6 TELEFONE
7 ALIMENT.
8 TOT.DESP.
9 SALDO
COMANDO ?<EN,AL,CA,DX,EX,IM,SA>
"L"
```

Após o preenchimento desta coluna, digite o comando desejado.

EN — entrada de dados
AL — alteração
CA — cálculo do saldo do mês
DX — move a planilha para a direita
EX — move a planilha para a esquerda
IM — imprime as despesas do mês
SA — grava o programa e os dados da planilha

Para a ENTRADA DE DADOS (linha 6Ø a 22Ø) digite EN e **NEW LINE** e depois o mês. Digite apenas os valores da entrada do mês e as despesas (luz, água, telefone, alimentação). Para digitar as outras células basta apertar a tecla **NEW LINE**, pois elas serão preenchidas na rotina de cálculo.

```
  CONTAS        JAN        FEV
1 ENTRADA      100000
2 SALD.ANT.
3 TOT.ENTR.
4 LUZ            5000
5 AGUA           4000
6 TELEFONE       5000
7 ALIMENT.      20000
8 TOT.DESP.
9 SALDO
COMANDO ?<EN,AL,CA,DX,EX,IM,SA>
"L"
```

Caso haja alguma alteração a ser feita (linha 41Ø a 53Ø) digite AL e N.L., em seguida o mês em que será feita a alteração e a linha a ser alterada. Depois, introduza o novo valor desta célula. Por exemplo: mês 1, linha 6 para 6ØØØ.

```
  CONTAS        JAN        FEV
1 ENTRADA      100000
2 SALD.ANT.
3 TOT.ENTR.
4 LUZ            5000
5 AGUA           4000
6 TELEFONE       5000
7 ALIMENT.      20000
8 TOT.DESP.
9 SALDO
MES= 1 LINHA= 6
"6000"
```

Sua planilha ficará assim:

```
  CONTAS        JAN        FEV
1 ENTRADA      100000
2 SALD.ANT.
3 TOT.ENTR.
4 LUZ            5000
5 AGUA           4000
6 TELEFONE       6000
7 ALIMENT.      20000
8 TOT.DESP.
9 SALDO
COMANDO ?<EN,AL,CA,DX,EX,IM,SA>
"L"
```

Quando todos os dados estiverem corretos, digite CA e N.L. Esta rotina de CÁLCULO (linha 72Ø a 84Ø) coloca o saldo do mês anterior no saldo anterior deste mês (linha 76Ø); soma a entrada do mês com o saldo anterior, colocando o resultado no total da entrada (linha 77Ø); calcula o total das despesas (linha 78Ø) e calcula o saldo do mês (linha 79Ø).

Digite o programa anexo e verifique o funcionamento deste Minicalc que apresentamos. Mas digite exatamente como a listagem se apresenta, sem alterar nomes ou números, caso contrário a Soma Sintática não será a mesma (o programa Soma Sintática deve ser colocado na **RAMTOP**, *portanto antes de digitar o programa).**

MEMÓRIA OCUPADA:
3.1Ø8 bytes

SOMA SINTÁTICA:
56.32Ø

** Veja Microhobby 5 ou X .*

```
  10 DIM U$(13,9,9)
  20 DIM U(13,9)
  30 LET C=0
  40 REM EN
  50 GOTO 130
  60 PRINT AT 21,0;"MES=
"
  70 INPUT M
  75 IF M>12 OR M<1 THEN GOTO 70
  80 LET C=M
  90 PRINT AT 21,5;M
 100 FOR F=1 TO 9
 110 LET U$(M+1,F)=" "
 120 NEXT F
 130 FOR D=1 TO 9
 140 GOSUB 850
 150 PRINT AT 21,0;"ENTRADA DE D
ADOS ";C;",";D
 160 INPUT H$
 170 IF CODE H$<=27 OR CODE H$>=
38 THEN LET U(C+1,D)=0
 180 IF CODE H$>27 AND CODE H$<3
8 THEN LET U(C+1,D)=VAL H$
 190 LET N=LEN H$
 200 IF C>0 THEN LET U$(C+1,D,10
-N TO )=H$
 210 IF C=0 THEN LET U$(C+1,D)=H
$
 220 NEXT D
 230 GOSUB 850
 240 PRINT AT 21,0;"COMANDO ?<EN
,AL,CP,DX,EX,IM,SA>"
 250 INPUT H$
 260 IF H$="DX" THEN GOTO 330
 270 IF H$="EX" THEN GOTO 370
 280 IF H$="CA" THEN GOTO 720
 290 IF H$="AL" THEN GOTO 410
 300 IF H$="EN" THEN GOTO 60
 310 IF H$="IM" THEN GOTO 540
 320 IF H$="SA" THEN GOTO 650
 325 GOTO 250
 330 REM DX
 340 LET M=M+1
 350 IF M>11 THEN LET M=11
 360 GOTO 230
 370 REM EX
 380 LET M=M-1
 390 IF M<1 THEN LET M=1
 400 GOTO 230
 410 REM AL
 420 PRINT AT 21,0;"MES=
"
 430 INPUT M
 435 IF M>12 OR M<1 THEN GOTO 43
0
 440 PRINT AT 21,5;M;" LINHA= ";
 450 INPUT Z
 455 IF Z>9 OR Z<1 THEN GOTO 450
 460 PRINT Z
 470 INPUT H$
 480 LET U$(M+1,Z)=" "
 490 IF CODE H$<=27 OR CODE H$>=
38 THEN LET U(M+1,Z)=0
 500 IF CODE H$>27 AND CODE H$<3
8 THEN LET U(M+1,Z)=VAL H$
 510 LET N=LEN H$
 520 LET U$(M+1,Z,10-N TO )=H$
 530 GOTO 230
 540 REM IM
 550 LET B$=" JANEIRO FEVEREIRO
 MARCO    ABRIL    MAIO     JUNH
O   JULHO   AGOSTO  SETEMBRO
OUTUBRO  NOVEMBRO DEZEMBRO"
 560 PRINT AT 21,0;"QUAL O MES ?
"
 570 INPUT M
 575 IF M>12 OR M<1 THEN GOTO 57
0
 580 PRINT AT 21,13;M
 590 LPRINT "  CONTAS",B$((M-1)*
9+1 TO (M*9))
 600 LPRINT
 610 FOR D=1 TO 9
 620 LPRINT U$(1,D);" .....";U$(
M+1,D)
 630 NEXT D
 640 GOTO 230
 650 REM SA
 660 PRINT AT 21,0;"PREPARE O GR
AVADOR E APERTE N.L."
 670 PAUSE 3000
 680 PRINT AT 21,0;"APERTE O REC
DO GRAVADOR
"
 690 PAUSE 150
 700 SAVE "MINICALC"
 710 GOTO 230
 720 REM CA
 730 PRINT AT 21,0;"QUAL O MES ?
"
 735 INPUT M
 740 IF M>12 OR M<1 THEN GOTO 73
5
 745 LET C=M
 750 FAST
 760 LET U(C+1,2)=U(C,9)
 770 LET U(C+1,3)=U(C+1,1)+U(C+1
,2)
 780 LET U(C+1,8)=U(C+1,4)+U(C+1
,5)+U(C+1,6)+U(C+1,7)
 790 LET U(C+1,9)=U(C+1,3)-U(C+1
,8)
 800 FOR F=1 TO 9
 810 LET N=LEN STR$ U(C+1,F)
 815 LET U$(C+1,F)=" "
 820 LET U$(C+1,F,10-N TO )=STR$
U(C+1,F)
 830 NEXT F
 840 GOTO 230
 850 REM TE
 860 FAST
 870 LET B$="JANFEVMARABRMAIJUNJ
ULAGOSETOUTNOVDEZ"
 880 LET A$="      CONTAS
                    "
 890 IF C=0 THEN LET M=1
 900 IF M=12 THEN LET M=11
 910 LET A$(16 TO 18)=B$(M*3-2 T
O M*3)
 920 LET A$(26 TO 28)=B$(M*3+1 T
O M*3+3)
 930 CLS
 940 LET B$="■:----------:--------
--:----------:"
 950 LET C$=":"
 960 PRINT A$
 970 FOR Z=1 TO 9
 980 PRINT B$
 990 PRINT CHR$ (Z+156);C$
1000 NEXT Z
1010 PRINT B$
1020 FOR F=1 TO 9
1030 PRINT AT F*2,2;U$(1,F);AT F
*2,12;U$(M+1,F);AT F*2,22;U$(M+2
,F)
1040 NEXT F
1050 SLOW
1060 RETURN
```

SUB-ROTINAS DO PROGRAMA

linha 1Ø	**– dimensiona a matriz da planilha**
linha 2Ø	**– dimensiona a matriz para os valores numéricos da planilha.**
linha 3Ø a 5Ø	**– entrada de dados na coluna "Ø" (Contas)**
linha 6Ø a 22Ø	**– entrada de dados por mês (colunas 1 a 12)**
linha 23Ø a 325	**– controle do comando a ser acionado**
linha 33Ø a 36Ø	**– move a planilha para a direita**
linha 37Ø a 4ØØ	**– move a planilha para a esquerda**
linha 41Ø a 53Ø	**– alteração de uma célula da planilha**
linha 54Ø a 64Ø	**– imprime a relação de despesas de um mês**
linha 65Ø a 71Ø	**– grava o programa e os dados da planilha**
linha 72Ø a 84Ø	**– calcula o saldo do mês**
linha 85Ø a 1Ø6Ø	**– elabora a formatação da tela**

	CONTAS	JAN.	FEV.
1	ENTRADA	100000	
2	SALD.ANT.	0	
3	TOT.ENTR.	100000	
4	LUZ	5000	
5	AGUA	4000	
6	TELEFONE	6000	
7	ALIMENT.	20000	
8	TOT.DESP.	35000	
9	SALDO	65000	

COMANDO ?<EN,AL,CA,DX,EX,IM,SA>
"■"

Os comandos DX e EX movem a planilha para a direita ou para a esquerda, possibilitando uma visualização global da planilha.

Se você possui uma impressora e deseja listar a relação de despesas do mês, utilize o comando de IMPRESSÃO (linha 54Ø a 64Ø).

CONTAS		JANEIRO
ENTRADA	.....	100000
SALD.ANT.	.....	0
TOT.ENTR.	.....	100000
LUZ	.....	5000
AGUA	.....	4000
TELEFONE	.....	6000
ALIMENT.	.....	20000
TOT.DESP.	.....	35000
SALDO	.....	65000

Para a gravação do programa, bem como os dados da planilha (linha 65Ø a 84Ø) digite SA e NEW LINE. Portanto, quando você recuperar o programa gravado numa fita cassete, através do comando LOAD " " e for manusear os dados gravados, NÃO UTILIZE o comando RUN, caso contrário, ele apagará os dados já armazenados.

Agora que você conhece o funcionamento do *Minicalc,* utilize sua imaginação e crie novos *Minicalcs.* Sugestão: crie um comando que possibilite o movimento da planilha para cima e para baixo.

Experimente digitar estes dados para o mês de fevereiro:

	CONTAS	FEV	MAR
1	ENTRADA	200000	
2	SALD.ANT.		
3	TOT.ENTR.		
4	LUZ	3000	
5	AGUA	5000	
6	TELEFONE	1000	
7	ALIMENT.	20000	
8	TOT.DESP.		
9	SALDO		

COMANDO ?<EN,AL,CA,DX,EX,IM,SA>
"■"

Depois faça os cálculos com o comando "CA" e a seguir mova a planilha para a esquerda utilizando a rotina "EX".

	CONTAS	JAN	FEV
1	ENTRADA	100000	200000
2	SALD.ANT.	0	65000
3	TOT.ENTR.	100000	265000
4	LUZ	5000	3000
5	AGUA	4000	5000
6	TELEFONE	6000	1000
7	ALIMENT.	20000	20000
8	TOT.DESP.	35000	29000
9	SALDO	65000	236000

COMANDO ?<EN,AL,CA,DX,EX,IM,SA>
"■"

Repare que o saldo do mês de janeiro foi colocado durante a rotina de cálculo no saldo anterior do mês de fevereiro. ○

COMPUTADORES NO RALLY

O III Rally Universitário Hermes Macedo, encerrado dia 10 de dezembro foi talvez o mais importante evento automobilístico desde 1980, quando recomeçaram as competições para estreantes e novatos suspensas em 1976.

Nestes últimos anos, mais e mais computadores têm se aliado ao homem para auxiliá-lo nas mais diversas atividades. A sua dinâmica tem sido explorada desde as profundezas do mar aos confins do espaço e seu trabalho tem sido reconhecido como indispensável pelo mundo todo.

A Microdigital esteve presente no III Rally patrocinando dois carros: o número 200, pilotado por Marcelo Chamie (o Lufão) e navegado por Tácito Mistrorigo de Almeida (o Tito), que correram como estreantes apenas na terceira etapa e embora tendo problemas mecânicos, obtiveram boa colocação. Mais sorte porém teve o carro número 381, pilotado por Francisco Rodes Fauss (o Chico) e navegado por Francisco Loschiavo Neto (o Chiquinho), que participaram como veteranos e que venceram a primeira etapa do Rally (Santos); foram 2º colocado na segunda etapa (Campinas) e na IV etapa (Sorocaba) e chegaram em 13º na terceira etapa (São José dos Campos), ficando classificados em segundo lugar na contagem geral dos pontos.

Contando apenas com competidores universitários, este Rally quebrou o recorde mundial de participantes: mais de 350, dos quais boa parte se utilizou das vantagens de se ter um computador ou calculadora a bordo para auxiliar nos cálculos de navegação com rapidez e precisão durante as provas. Computadores como o TK-85 tiveram papel decisivo para se alcançar boas colocações, na opinião de navegadores como o Tito e o Chiquinho que, insatisfeito como segundo colocado, promete ganhar disparado na próxima corrida, contando mais uma vez com a ajuda de um TK-85 da Microdigital.

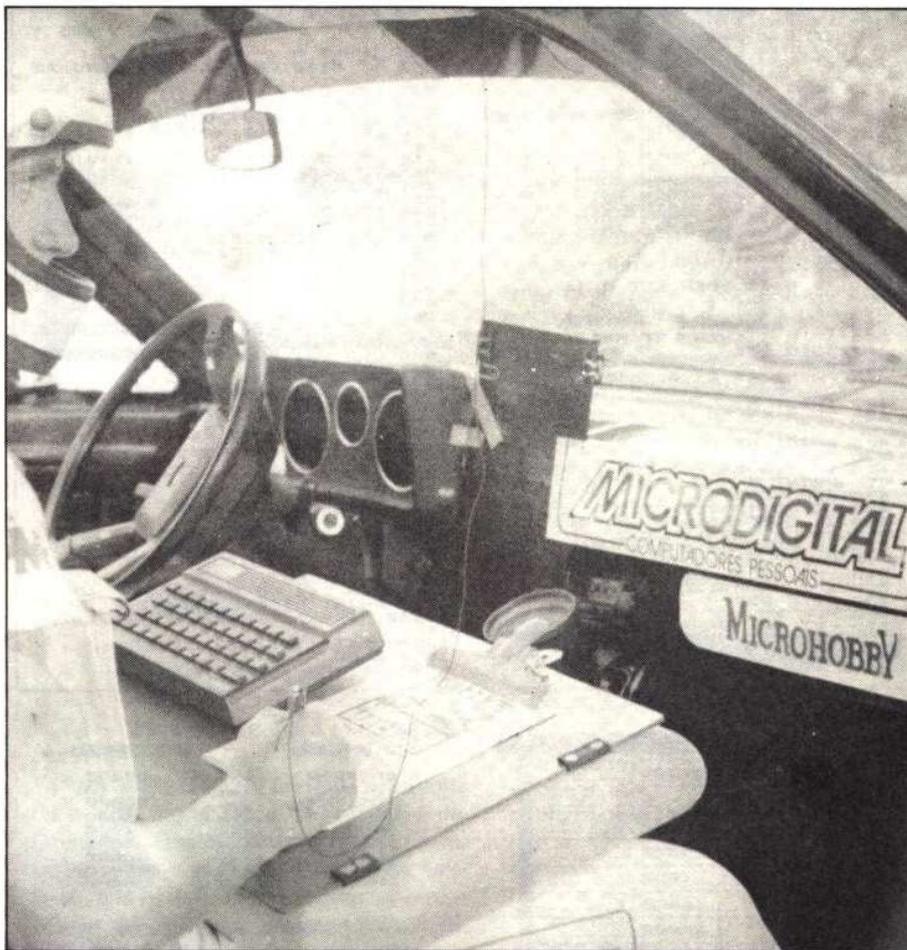
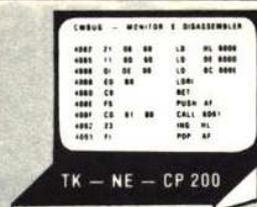



FOTO: Walter Morais

Show a parte, porém foi a maneira de instalar os computadores nos carros tanto elétrica quanto mecanicamente. Apesar de ter de suportar trepidações durante cerca de 300 km por 6 horas, havia até carro equipado com TRS-80 da Radio Shack! A equipe patrocinado pela Microdigital (as duplas Chico-Chiquinho e Lufão-Tito e o técnico em Hard/Soft, Tanios Hamzo) optou porém pelo conjunto micro-TV, que apesar de dar mais trabalho, ficava mais portátil e maleável. Quanto à parte eletrônica (os cinco integrantes da equipe são estudantes de engenharia eletrônica na E.E. Mauá) o "know-how" desenvolvido será analisado na sessão HOBBY do próximo número desta revista, e esperamos que lhe seja útil, mesmo que você não esteja pensando em tomar parte em um Rally, mas que possa querer instalar seu computador no carro.○

TK-FILE
GERENCIADOR DE BANCO DE DADOS PARA A LINHA TK E COMPATÍVEIS!
Programa com alta velocidade de processamento, pelo emprego de rotinas em código de máquina.
Com ele você terá um versátil arquivo, com grande liberdade para definir os itens desejados, formar fichas, formatar e imprimir os relatórios classificados, selecionar, alterar, incluir e excluir dados e armazenar seu arquivo em fita cassete.
Acompanham completas instruções para operação do programa.
Para adquiri-lo via correio, basta enviar carta com seu nome e endereço completos anexando cheque no valor de Cr$ 11.790,00, nominal à MULTISOFT INFORMÁTICA LTDA. - Caixa Postal 54121 - CEP 01296 - SÃO PAULO - SP
MULTISOFT
ENTREGA IMEDIATA PARA TODO BRASIL
Fitas com certificado de garantia
40 programas disponíveis
solicite folheto

CWBUG
MONITOR E DISASSEMBLER
LANÇAMENTO CWB
TK — NE — CP 200
— Fita cassete
— Manual de instruções
— Certificado de garantia
* 4 kbytes de programa totalmente em linguagem de máquina.
* Decodifica blocos de instruções em linguagem de máquina para mnemônicos assembly Z-80 padrão ZILOG.
* Possui 23 comandos monitores para ler, escrever, transferir, apagar, executar, gravar, carregar blocos de programas em linguagem de máquina.
* Lê registros da CPU e endereços da ROM e RAM com saída hexadecimal e alfanumérica.
* Permite operação simultânea com outros programas em BASIC ou linguagem de máquina.
* Ferramenta indispensável para estudo, desenvolvimento e edição de programas em linguagem assembly compatível com os micro-computadores TK-82 / TK-83 / TK-85 NE-Z 8000 e CP 200 - 16 K.
— Desejo adquirir o programa CWBUG. Estou anexando cheque Nº ________ Banco Nº ________ , nominal à CWB MICROCOMPUTADORES LTDA, no valor de CR$ 8.500,00.
OBS: As despesas de remessa estão incluídas no preço.
NOME: ________
ENDEREÇO: ________
CIDADE ________ ESTADO ________ CEP ________
CWB
CWB MICROCOMPUTADORES LTDA
CAIXA POSTAL - 3447
80.000 - CURITIBA - PR

NOVIDADES

O MICRO ROBÔ QUE JOGA XADREZ

Uma novidade bastante interessante, lançada na -- ainda comentada -- feira da informática do último ano, é o simpático micro robô KALT-400. O robô é formado por um braço mecânico com seis eixos, que se movimentam em ângulos que variam de 0° a 180°. Ele pode ser utilizado em diversas atividades; desde o auxílio a deficientes físicos até ao lado de um solitário jogador de xadrez, como um verdadeiro parceiro.

O KALT-400 atende a comandos programados em qualquer microcomputador (inclusive os compatíveis com o TK-83), como também comandos em qualquer linguagem. No BASIC, foi realizada uma experiência que resultou no seguinte miniprograma: 10 POKE X,Y, onde *X* define o eixo e *Y* a posição do eixo selecionado.

O BASIC PRESENTE NO INTERIOR DE SÃO PAULO

Marília, cidade do interior de São Paulo, preocupando-se com a necessidade emergente por informações na área da Informática, inaugurou o CIM — Centro de Informática Marília.

O Centro oferecerá cursos de BASIC, a linguagem mais usual e simples de computação, em três níveis. O primeiro e o segundo, atingirão as crianças a partir dos cinco anos de idade e trabalharão com elas, no sentido de desenvolverem a sua criatividade, através do raciocínio lógico e crítico, dentro de uma visão humanística. O terceiro nível, destina-se ao público jovem e adulto, oferecendo-lhes os principais conceitos relacionados ao uso do microcomputador na linguagem BASIC, evidente!

UMA MÁQUINA DE ENDEREÇAR

Foi lançado em São Paulo pela Novel Print (empresa fabricante de etiquetas e fitas) uma máquina de endereçamento postal automático. O equipamento, apresentado na última feira da Informática, denomina-se **Tickopress 19 SD/MD** e foi idealizada com o objetivo de efetuar um

endereçamento automático em todo tipo de correspondência, desde envelopes até revistas. O processo de endereçamento é realizado com o empilhamento dos envelopes na parte anterior da máquina que os separa por unidades, aplicando as etiquetas sempre na mesma disposição, alcançando uma velocidade média de até 70 aplicações por minuto.

FITAS COLORIDAS PARA IMPRESSORA

Investindo grande potencial na ampliação de suas empresas, o Grupo Machado, formado pelas empresas Data Ribbon, Mr. Data e Datrix, um grupo fabricante de fitas magnéticas e disquetes, resolveu partir, neste segundo semestre, para um ritmo mais intenso de automação.

Com este objetivo, o *Grupo* está desenvolvendo em sua fábrica, um laboratório de análises e instalando máquinas de entintar em OCR e corte de filme de polietileno. Todo este investimento em máquinas, resultou no lançamento dos disketes **Datadisk,** das fitas **Qume, LGM 3287** (em 4 cores), **Grafix 80 e 100**, além de uma linha completa de fitas em cores.

OS JOGOS DA MULTISOFT

Os usuários de micros têm agora à sua disposição uma série de 16 jogos em fitas, lançados pela Multisoft. São fitas que agrupam de dois a três programas com 2 K de memória, e podem ser utilizados em microcomputadores como os TKs 82 e 83. Entre os jogos disponíveis para venda nos revendedores estão: "Invasores, Desafio Espacial, Bombardeio, Smag-Smag, Confronto, Minotauro, Laser, Limpeza Cósmica entre outros.

MICRODIGITAL NO RIO

Os usuários dos micros fabricados pela Microdigital, no Rio, já possuem um local para levar os seus aparelhos encrencados. A empresa inaugurou em Ipanema, na Rua Visconde de Pirajá, 414-conj.606, uma filial para prestar assistência técnica aos usuários de seus aparelhos. O

EQUAÇÕES ALGÉBRICAS E TRANSCENDENTAIS

Pedro Antonio Carlini P. de Souza

1. O MÉTODO

O método utilizado no programa denomina-se "Método de Bolzano ou de bissecção". É um processo simples de aproximação sucessiva para determinar raízes reais de uma equação algébrica ou transcendental. Partindo-se de um ponto '*x*' menor que a raiz procurada, o método incrementa '*x*', sucessivamente, por 'h', até que um intervalo de amplitude '*h*' seja encontrado, de modo que f(x) e f(x+h) tenham sinais opostos, isto é, $f(x).f(x+h) < \emptyset$.

Uma raiz então é detectada entre '*x*' e '*x+h*'. A partir daí, o passo '*h*' é reduzido à metade em todas as etapas seguintes, observando-se o seguinte:

a. Se $f(x).f(x+h) > \emptyset$, na etapa seguinte usar-se-á o intervalo anterior reduzido à metade, com valor de 'x' incrementado pelo valor de 'h' (dessa etapa anterior), isto é, daquela em que foi obtido $f(x).f(x+h) > \emptyset$ [sinal (+)]; e

b. Se $f(x).f(x+h) < \emptyset$, na etapa seguinte usar-se-á o intervalo anterior reduzido à metade, porém com o valor de 'x' mantido da etapa anterior, isto é, daquela em que foi obtido: $f(x).f(x+h) < \emptyset$. [sinal (−)].

Desta forma, a raiz é sempre aproximada pela esquerda e em qualquer etapa, a amplitude do intervalo do passo anterior dá o limite superior de erro.

Encontrada uma raiz, pode-se partir de sua direita para localizar outra que porventura houver.

2. O PROGRAMA

a. Comandos BASIC

```
  10 REM PEDRO A. CARLINI
  15 REM "RAIZES DA EQUACAO"
  20 REM "METODO: BOLZANO"
  40 LET K=0
  50 LET L=0
  60 PRINT AT 5,1;"VOCE JA" INTR
ODUZIU A FUNCAO     DESEJADA NA L
INHA 600 ? (S/N)"
  70 INPUT U$
  80 IF U$="S" THEN GOTO 120
  90 CLS
 100 PRINT AT 5,1;"LISTE O PROGR
AMA E INTRODUZA A FUNCAO DESEJAD
A NA LINHA 600."
 110 STOP
 120 LET H=0.5
 130 CLS
 140 PRINT AT 5,1;"ENTRE COM UM
VALOR QUALQUER      MENOR QUE A R
AIZ PROCURADA."
 150 INPUT X
 160 LET K=K+1
 170 CLS
 180 GOSUB 600
 190 LET G=F
 200 PRINT AT 5,1;"X= ";X;AT 7,1
;"H= ";H;AT 9,1;"X+H= ";X+H
 210 LET X=X+H
 220 GOSUB 600
 230 PRINT AT 11,1;"F(X)= ";G;AT
13,1;"F(X+H)= ";F
 240 IF F=0 THEN GOTO 510
 250 IF G=0 THEN GOTO 530
 260 IF ABS (F-G)<=.0001 THEN GO
TO 530
 270 IF F*G>0 THEN GOTO 310
 280 PRINT AT 15,1;"SINAL (-)"
 290 LET L=1
 300 GOTO 320
 310 PRINT AT 15,1;"SINAL (+)"
 320 PRINT AT 17,1;"DIGITE <NEW
LINE>"
 330 PAUSE 9999
 340 POKE 16436,255
 350 IF NOT L=1 THEN GOTO 400
 360 IF F*G>0 THEN GOTO 380
 370 LET X=X-H
 380 LET H=H/2
 390 GOTO 160
 400 IF K<=10 THEN GOTO 160
 410 CLS
 420 PRINT AT 5,1;"APOS INCREMEN
TAR <X> DEZ VEZES, NAO ENCONTREI
 INDICIOS DE RAIZ."
 430 PRINT AT 9,1;"PARA DESISTIR
 DIGITE <P>."
 440 PRINT AT 11,1;"PARA TENTAR
OUTRO VALOR INICIAL DIGITE QUALQ
UER TECLA E NEWLINE"
 450 INPUT U$
 460 IF U$="P" THEN STOP
 470 CLEAR
 480 LET K=0
 490 LET L=0
 500 GOTO 120
 510 PRINT AT 17,1;"X= ";X;" E"
A SOLUCAO."
 520 STOP
 530 PRINT AT 17,1;"X= ";X-H;" E
" A SOLUCAO."
 540 STOP
 600 LET F=EXP X-3*X
 610 RETURN
 620 STOP
 630 SAVE "RAIZES DE EQUACAO"
 640 RUN
```

MEMÓRIA OCUPADA:
1556 bytes
SOMA SINTÁTICA:
21289

b. Variáveis utilizadas

- **X:** valores das abscissas da função f(x);
- **H:** valores das amplitudes dos intervalos utilizados;
- **G:** valores para f(x);
- **F:** valores para f(x÷h); e
- **K, L e U$:** valores auxiliares para controle das mensagens pertinentes às diversas situações encontradas.

c. Observações

1) A função f(x) da qual se quer pesquisar o valor de 'x' tal que $f(x) = \emptyset$, deve ser, previamente, definida na subrotina 6ØØ. Caso isto seja esquecido pelo pesquisador, o programa irá lembrá-lo desse fato;

2) O intervalo '*h*' está, inicialmente, fixado em 0,5 de amplitude.

3) Está previsto o caso onde um dos valores assumidos por '*x*' ou '*x+h*' venha a se constituir em uma raiz exata da equação.

4) Caso o valor inicial escolhido para '*x*' não permita, após dez etapas, encontrar um intervalo [x, x+h] tal que $f(x).f(x+h) < \emptyset$, o pesquisador será consultado se desiste da pesquisa, ou se quer atribuir outro valor inicial para *X*, reiniciando o processo iterativo;

5) O programa pode ser utilizado num TK83, não sendo necessária a expansão de memória.

6) O cálculo da raiz está previsto com aproximação a menos de Ø,ØØØ1.

3. O TESTE

O programa foi testado com diversas funções e valores arbitrários iniciais para '*x*'.

Os comandos "basic" do parágrafo anterior apresentam para teste a equação $e^x - 3x = \emptyset$. Para tanto, definiu-se na linha 6ØØ a função:

```
600 LET F=EXP X-3*X
```

Partindo-se do valor inicial $x = \emptyset$, depois de 15 impressões de resultados, obtêm-se a solução. Reproduzimos a seguir alguns resultados:

a. Primeira impressão:

```
X= 0.6171875
H= .00390625
X+H= 0.62109375
F(X)= .0021446524
F(X+H)= -.0023188936
SINAL (-)
DIGITE <NEW LINE>
```

b. Segunda impressão:

```
X= 0.6171875
H= .0009765625
X+H= 0.61816406
F(X)= .0021446524
F(X+H)= .0010261093
SINAL (+)
DIGITE <NEW LINE>
```

c. Décima quinta impressão:

```
X= 0.61901856
H= .000061035156
X+H= 0.61907959
F(X)= .000048836693
F(X+H)= -.000020916574

X= 0.61901856 E" A SOLUCAO.
```

O autor deste programa, é bacharel licenciado em matemática pela PUC-SP e atualmente dá aulas na área, na PUC-SP e nas Faculdades Oswaldo Cruz.

CHEGA DE PROBLEMAS!

APENAS Cr$ 18.000,00

TIG-LOADER

D L IN F

use Tig Loader.

TIG-LOADER possibilita:

- *a localização do ótimo volume do gravador, facilitando a operação LOAD.*
- *DUPLICAR qualquer programa, mesmo aqueles "fechados".*
- *carregar (LOAD) e DUPLICAR simultaneamente.*
- *gravar (SAVE) em 2 gravadores ao mesmo tempo.*
- *monitorar as operações LOAD, SAVE ou DUPLICAÇÃO através de fone.*
- *filtrar as interferências elétricas de baixa frequência, que são a causa da maioria dos problemas de LOAD/SAVE.*

NÃO CARREGUE ESTES PROGRAMAS!

V. não precisa mais carregar estes programas: eles estão prontos para uso, em CARTUCHO! Conecte o cartucho, ligue o micro e... só! Seis aplicativos à sua disposição, esperando V. comandar. SEM modificar seu micro. E a expansão de memória é usada normalmente! Como lançamento, o TIGRE oferece: hi-speed, renumerador, apagador de linhas em bloco, soma sintática, disponibilidade de memória e rotinas de vídeo em um único cartucho! É ESPETACULAR!!!

ESCREVA PEDINDO INFORMAÇÕES

TIGRE
COM. DE EQUIP. P/COMPUTADORES LTDA.
Rua Correia Galvão, 224 - CEP 01547 - São Paulo

Envie seu pedido + cheque cruzado.
Prazo de entrega: 15 dias.
Despesas postais incluidas nos preços.

Atendemos somente por carta.
Solicite relação de programas em fita.

Leoni

SIM, *desejo receber o* **TIG-LOADER**, *para tanto, estou anexando o cheque n.º* ______ *no valor de Cr$* ______

NOME:			ENDEREÇO:	
CEP:	CIDADE:	ESTADO:	PROFISSÃO:	ASSINATURA:
IDADE:	MARCA DO MICROCOMPUTADOR:			

DUELO

Gustavo Egidio de Almeida

Você é **Kid Dedo Lerdo**, o melhor (e único) xerife da região onde fica **MicroTown.**

Lá, encontram-se Jonny Ringo, Jesse James, Billy The Kid, Sundance Kid, os irmãos Dalton, Kid Reagan e outros grandes matadores. Você veio à **MicroTown** para acabar com todos eles, e agora segue pela rua principal da cidade.

Cuidado! Em cada esquina pode haver alguém à sua espera, e, se assim for, você não poderá fugir ao **DUELO**!

O confronto será inevitável e só lhe restarão duas opções. . . *Matar* . . . ou *Morrer*!

Aguarde o momento certo e. . . *"Fogo!!"* (Figura 1).

PONTOS: 200

Aquele que disparar primeiro, ficará para contar a história . . .

Eu sinto pena de você, pois o temível e esquizofrênico Kid Reagan, o maior de todos os matadores, o odeia! Decidido a liquidar você, trouxe à **MicroTown** toda a sua quadrilha e você só sairá da cidade num caixão!

Portanto, se você quiser continuar vivo, continue matando!

Tome cuidado com sua consciência, pois você é uma pessoa honesta e se disparar antes do momento . . . "Fogo!", irá punir a si mesmo (Figura 2).

PONTOS: 200

VOCE DESOBEDECEU REGRAS

Contagem:

— A cada passo percorrido → 5 pontos
— Caso vença o **Duelo** → 50 pontos
— Desobedecendo as regras → –5 pontos

Para mover, (O) use a tecla 7
Para atirar use a tecla Ø
Inimigo — (")

Muita atenção no momento da digitação!

Cuidado para não digitar comandos ou funções, letra por letra.

Funções, como por exemplo INKEY$, devem ser digitadas da seguinte maneira: SHIFT NEW LINE tecla que contém a função.

Caracteres gráficos usados:

" ▩ " – SHIFT 9 e SHIFT H
" ■ " – SHIFT 9 e SHIFT SPACE
" ⊡ " – SHIFT 9 e SHIFT B

Observação: Para alterar o grau de dificuldade do jogo, devemos listar a linha 35Ø através do comando LIST 35Ø e NEW LINE, depois usar SHIFT EDIT.

Feito isso, é só mover o cursor para a direita através do comando SHIFT 8 repetindo a operação até que o cursor pare de se mover.

Agora, delete o último caractere, digitando SHIFT RUBOUT. Digite uma das letras abaixo:

B
A
R

Pronto, está feita a alteração. Experimente fazê-las e veja o que acontece em cada uma delas.

Como você pode notar, na listagem do programa, são utilizadas funções pouco usuais para computadores de 16 K ou mais de memória, mas muito usadas em programas com no máximo 2 K de RAM. Funções "estranhas" como VAL, mas muito importantes porque economizam bytes, diminuindo significativamente a

MEMÓRIA OCUPADA:
922 bytes
SOMA SINTÁTICA:
57 Ø77

memória do programa.

Por exemplo:

Quando fazemos:

1Ø LET C = 1Ø NEW LINE,

ocupamos 16 bytes.

Agora, se ao invés disso, fizermos:

1Ø LET C = VAL "1Ø" NEW LINE

ocuparemos 13 bytes.

Você pode constatar isso através do comando: PRINT PEEK 16396 + 256 * PEEK 16397 – 165Ø9.

AGUARDE:
COMO ECONOMIZAR MEMÓRIA
NO TK, NA SEÇÃO **DICAS.**

```
   2 CLS
   3 LET K=VAL "1"
   4 LET R=VAL "10"
   5 LET A=VAL "8"
   6 LET P=K-K
   7 LET B=VAL "4"
   8 LET S=R-K
   9 LET D=VAL "50"
  20 FOR F=P TO A STEP K+K
  25 PRINT AT F,P;"██   ██",,"
█    █"
  40 NEXT F
  50 PRINT AT K,R;"PONTOS:"
  60 PRINT AT S,B;"O";AT S,B;" "
  70 IF INKEY$="7" THEN LET S=S-
(K+K)
  75 IF INKEY$="7" THEN LET P=P+
R
  77 IF S<K THEN LET S=R-K
  80 LET L=INT (RND*B)
  90 IF L=K THEN GOTO VAL "200"
 150 GOTO D+R
 200 PRINT AT S,B;"O"
 205 LET C=INT (RND*(K+K))
 220 IF C=K-K THEN LET C=A-K
 230 FOR H=K TO R
 240 PRINT AT S-(K+K),C;"█";AT S
-(K+K),C;""""
 250 NEXT H
 255 PRINT AT K,R+A;P
 260 FOR F=K TO B-K
 270 PRINT AT S-(K+K),C;"""";AT
S-(K+K),C;" "
 275 LET C=C+(C<B)-(C>B)
 280 NEXT F
 290 PRINT AT S-(K+K),C;""""
 300 PRINT AT R,R;"████████"
 310 FOR F=K TO INT (RND*D)+D
 320 IF INKEY$="0" THEN GOTO VAL
"400"
 330 NEXT F
 340 PRINT AT R,R;"████████"
 350 FOR F=K TO INT (RND*B-K)+K
 360 NEXT F
 365 IF INKEY$="0" THEN GOTO VAL
"500"
 375 PRINT AT S,C;".";AT S,C;"X"
 385 PRINT AT R,R;"TECLE P"
 387 IF INKEY$="P" THEN RUN
 390 GOTO VAL "375"
 400 PRINT AT R,A;"VOCE DESOBEDE
CEU REGRAS"
 410 PAUSE VAL "200"
 425 PRINT AT S-(K+K),C;" "
 430 LET P=P-(B+K)
 445 PRINT AT R,A;"
              "
 450 GOTO D
 500 FOR F=K TO R
 510 PRINT AT S-(K+K),C;".";AT S
-(K+K),C;"X"
 520 NEXT F
 530 LET P=P+D
 535 PRINT AT S-(K+K),C;" "
 540 GOTO VAL "445"
```

14 programas para meditação

Flavio Rossini

Este pequeno artigo simplesmente apresenta alguns programas escritos em Basic, cujo efeito deveria ser analisado para "meditação" por todos que já tenham um bom conhecimento de Basic-TK. Comecemos brincando com a tela. . .

```
PROGRAMA 1

10 SLOW
20 FAST
30 RUN
```

```
PROGRAMA 2

10 FAST
20 SLOW
30 RUN
```

```
PROGRAMA 3

10 PRINT AT 21,31;" "
20 PRINT "HARMÔNIA"
```

```
PROGRAMA 4

 5 SLOW
10 FOR I=0 TO 21
20 FOR J=0 TO 31
30 PRINT AT I,J;"▒"
40 PLOT I,J
50 NEXT J
60 NEXT I
```

Agora, vamos à precisão numérica do TK:

```
PROGRAMA 5

10 LET X=1234567890
20 PRINT X
```

```
PROGRAMA 6

10 LET X=0.00001
20 LPRINT X
```

```
PROGRAMA 7

10 LET X=-4
20 PRINT X**2
```

Como você explica os números aleatórios gerados pelo próximo programa?

```
PROGRAMA 8

10 SLOW
20 RAND
30 LET X=INT (100*RND)
40 SCROLL
50 PRINT X
60 RUN
```

Execute o programa a seguir e, no final, faça um LIST do mesmo . . .

```
PROGRAMA 9

1 PRINT "LENNON"
2 PRINT "MCCARTNEY"
3 POKE 16510,2
4 GOTO 2
```

Agora, com o auxílio de um relógio, cronometre o tempo de execução dos próximos dois programas. . . Tente explicar a diferença!

```
PROGRAMA 10

10 SLOW
20 FOR Y=0 TO 9
30 FOR X=0 TO 9
40 PLOT X,Y
50 NEXT X
60 NEXT Y
```

```
PROGRAMA 11

10 SLOW
20 FOR Y=34 TO 43
30 FOR X=54 TO 63
40 PLOT X,Y
50 NEXT X
60 NEXT Y
```

Agora, tente explicar a "assimetria" que ocorre nesta figura após o cruzamento das duas "linhas":

```
PROGRAMA 12

10 SLOW
20 FOR I=0 TO 21
30 PRINT TAB I;"■";TAB (21-I);
"■"
40 NEXT I
```

Utilizando as teclas 5, 6, 7, 8 ou o joystick, movimente o "ponto piscante" gerado pelo próximo programa *para cima* (tecla 7). O que ocorre quando você tentar sair da tela? Isto acontece em todas as direções?

```
PROGRAMA 13

10 SLOW
20 LET X=10
30 LET Y=20
40 LET XN=X
50 LET YN=Y
60 PLOT XN,YN
65 LET XN=X+(INKEY$="8")-(INKE
Y$="5")
70 LET YN=Y+(INKEY$="7")-(INKE
Y$="6")
75 UNPLOT X,Y
80 LET X=XN
85 LET Y=YN
90 GOTO 60
```

E, para finalizar, um pouco de lógica. . .

```
PROGRAMA 14

 10 SLOW
 20 LET A=1
 30 PRINT (NOT A)
 40 PRINT (A>3)
 50 PRINT (A=1)
 60 PRINT (A<2 AND A>3)
 70 PRINT (A>=1 OR A>-1)
 80 DIM A$(10)
 90 LET A$="BEATLES"
100 LET A=LEN A$
110 PRINT A$,A
```

PAREDÃO E SÃO FRANCISCO

Para aqueles que têm um computador de 2 k de RAM e não possuem expansão, a **Microhobby** apresenta, este mês, dois programas divertidos.

O primeiro é o *"São Francisco"*, uma cidade dos Estados Unidos que já sofreu vários abalos sísmicos(por estar construída sobre uma falha geológica)e que agora, (no nosso contexto) está sendo abalada por um tremor de terra e sendo destruída gradativamente. Pouco a pouco vão surgindo buracos (GRAPHIC SHIFT H) devido ao terremoto e você, o "O" representado na tela, como o último dos habitantes, deve lutar pela sobrevivência mas nunca abandonar a sua cidade.

Utilize as teclas 5, 6, 7 e 8 para locomover-se à esquerda, direita, descer e subir; ou do Joystick, escapando dos buracos que o ameaçam.

O jogo termina quando você cair em um deles ou sair das delimitações da cidade.

O segundo é o "PAREDÃO". Para quem nunca jogou *paredão* e quer saber como jogar, aqui está sua oportunidade!

Você (GRAPHIC SHIFT 5) deve rebater a bolinha (O) que bate na parede (na tela, localizada à esquerda), utilizando-se das teclas 6 para descer e 7 para subir. Quando você efetuar um total de três jogadas, na tela aparecerá o total de pontos conseguidos.

Conhecendo estes dois jogos, mãos à obra!

Introduza os programas tomando cuidado para nunca digitar as funções RND, PEEK, INKEY$, INT, VAL, PI, letra por letra — isso deve ser feito com o cursor em **F**. Por exemplo: para obter a função RND, tecle SHIFT NEW LINE T.

Para conferir os programas, utilize-se da Soma Sintática (publicada na edição número 5 de **Microhobby**). Para isto, coloque inicialmente o programa da Soma Sintática e "dê" RAMTOP. Feito isto, digite os programas exatamente conforme as coordenadas dadas, caso contrário, as modificações efetuadas por você nos programas, alterarão o resultado da Soma Sintática.

SÃO FRANCISCO

```
  10 PRINT "▛▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▜"
  20 FOR N=1 TO 14
  30 PRINT "▌                              ▐"
  40 NEXT N
  50 PRINT "▙▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▟"
  60 LET P=16924
  70 LET S=0
  80 POKE P,52
  90 PRINT AT 20,0;"PLACAR=";S
 100 PRINT AT 1+(RND*13),1+(RND*
11);"▒"
 110 IF PEEK P<>52 THEN STOP
 120 POKE P,0
 130 LET S=S+1
 140 LET P=P+(INKEY$="8")-(INKEY
$="5")+33*((INKEY$="6")-(INKEY$=
"7"))
 150 IF PEEK P<>0 THEN STOP
 160 GOTO 80
```

MEMÓRIA OCUPADA:
550 bytes
SOMA SINTÁTICA:
21409

PAREDÃO

```
   5 LET T=VAL "0"
  10 LET S=VAL "1"
  20 LET B=VAL "11"
  30 LET C=B
  40 LET A=VAL "12"
  45 PRINT "JOGO ";S
  50 PRINT AT B,VAL "13";" "
  60 PRINT AT C,A;" "
  65 IF A=VAL "0" THEN LET C=INT
 (RND*5)+10
  70 LET A=A-PI/PI
  80 LET B=B-(INKEY$="7")+(INKEY
$="6")
 100 PRINT AT B,VAL "13";"▌"
 105 LET A=A-PI/PI
 110 PRINT AT C,A;"O"
 120 IF A=VAL "-12" AND B=C THEN
 LET A=VAL "12"
 125 LET T=T+PI/PI
 130 IF A=VAL "-20" THEN GOTO VA
L "200"
 150 GOTO VAL "50"
 200 LET S=S+PI/PI
 205 CLS
 210 IF S<VAL "4" THEN GOTO VAL
"20"
 220 PRINT "SEU PLACAR = ";T
```

MEMÓRIA OCUPADA:
360 bytes
SOMA SINTÁTICA:
23565

O Instan-TK e a amnésia em micros.

Tanios Hamzo

Não há nada mais medonho para um digitador do que a idéia de ver um programa quilométrico (aqueles com 15,9 k Bytes) evaporar, antes de tê-lo gravado em fita.

Alguns computadores mais recentes possuem um tipo de memória que jamais esquece, a não ser propositadamente. Os TKs e tantos outros computadores pessoais têm o que se chama de "memória volátil", ou seja: interrompendo-se a alimentação elétrica, mesmo por muito pouco tempo, a memória "esquece" para sempre seu conteúdo.

Para espantar este "fantasma da amnésia" de uma vez por todas, costuma-se usar um sistema conhecido por NO BREAK, que não interrompe a alimentação elétrica ao micro, mesmo que haja falta de energia elétrica na rede, ou, num caso comum (pelo menos para mim) onde o usuário, muito excitado durante um jogo, chute sem querer o fio da tomada.

Em sistemas de responsabilidade, a instalação de um NO BREAK é crucial. O TK, que pode ter aplicações diversas, pode ter diferentes motivos para conservar sua memória por prazos igualmente variados.

Segundo minha concepção, são três as categorias de NO BREAK:

1. INSTANTÂNEA: Quando se quer que a memória não se perca por alguns instantes, enquanto se troca de tomada; quando houver instalação elétrica intermitente (com bruscas variações de tensão ou em tomadas, extensões ou "benjamins" com mau contato, evidenciado quando se mexe neles para ligar um gravador, por exemplo) ou ainda quando aquele seu querido priminho vem te visitar e quer descobrir o que acontece se o computador for desligado quando você está usando. A memória será mantida por alguns minutos.

2. PORTÁTIL: Quando se quer que a memória seja mantida, enquanto o computador é transportado de sua casa até a casa de um colega; da escola até o clube ou do escritório até um cliente. A memória poderá ser mantida por algumas horas.

3. PERPÉTUA: Em sistemas realmente NO BREAK, que não podem parar: alarmes residenciais, comerciais ou industriais; controle de processos; em softhouses e em ocasiões em que o micro deva funcionar indefinidamente sem interrupções, independente da disponibilidade da rede elétrica. A memória poderá ser mantida por dias.

Abordarei o primeiro e o segundo caso, que me parecem mais interessantes para o leitor, porém ficarei atento à correspondência pedindo complementação do artigo.

Chamei o projeto de INSTAN-TK e seu esquema elétrico pode ser visto na figura 1.

Todos estes componentes podem ser facilmente encontrados em casas do ramo. Atenção para a polaridade de diodos e do L.E.D., bem como a sua sensibilidade ao calor excessivo. Verifique as ligações antes de ligar o aparelho.

FUNCIONAMENTO DO CIRCUITO: Apesar de ser um circuito aparentemente simples em sua configuração física, há diversos detalhes dinâmicos acumulados em cada componente. Farei uma explanação do funcionamento conjunto, descrevendo a(s) função(ões) de cada parte.

Ao se ligar o plugue da fonte no circuito, através de J1 e este ao computador através de J2, observa-se:

Com a fonte ligada, haverá passagem de corrente para o computador através de D1, que está diretamente polarizado, sempre que a fonte estiver ligada. No entanto, quando ela não for mais capaz de suprir energia, ficará o diodo D1 inversamente polarizado, não conduzindo energia das pilhas para outro lugar senão para o computador. Com isso, ficarão com carga destinada exclusivamente para o computador, garantindo seu funcionamento. Além disto, este diodo melhora a atuação da fonte, aumentando sua retificação, sua impedância e ainda reduzindo em cerca de 0,6 volts sua tensão de saída, o que pode contribuir para minimizar o aquecimento do regulador interno de tensão do computador.

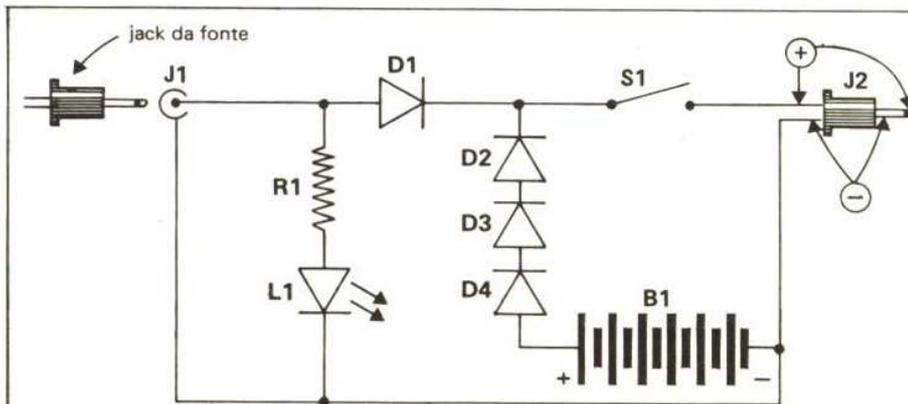



Figura 1: INSTAN-TK, esquema elétrico.

LISTA DE MATERIAL:

Quantidade	Componente/Tipo	Legenda
4	Diodo 1N4002	D1, D2, D3 e D4
1	L.E.D. de qualquer côr	L1
1	Interruptor de 1 pólo	S1
1	Jack fêmea	J1
1	Jack macho	J2
1	Resistor de 820 Ω, 1/4W	R1
6	Pilhas alcalinas de 1,5 V	B1
6/1	Suporte(s) para pilhas	—
1	Caixa plástica	—

Diversos: fio, solda, placa de circuito impresso ou ponte de terminais, etc.

O L.E.D. (diodo emissor de luz) L1 serve para se monitorar a atividade da fonte, apagando, quando ela não estiver operando e atestando seu funcionamento, quando estiver ligada. O resistor R1 limita o consumo do L.E.D. em cerca de 0,01 ampères.

O interruptor S1, além de proporcionar a interrupção da corrente, desligando o aparelho, também pode ser útil como um RESET, pois não tendo a inércia dos eletrolíticos da fonte, proporciona um rápido POWER-UP e POWER-DOWN.

Os diodos D2, D3 e D4 formam, em série uma diferença de potencial tal que os 9 Volts provenientes da associação das pilhas acabam se equiparando aos vales da flutuação da tensão da fonte. Estes vales podem chegar próximos a 7 volts, equilibrados com os 9 menos os 2 volts resultantes dos três diodos. No protótipo, a d.d.p. foi de 0,015 volts, insuficientes para provocar uma descarga nas pilhas (que aliás andam bem carinhas). Estes diodos também impedem que os picos de tensão da fonte polarizem reversamente as mesmas; fazem com que sejam mais lentamente consumidas, aumentando sua vida útil. A impedância adicional que estes diodos criam, também contribui para reduzir a corrente de descarga.

As pilhas substituem a energia da fonte automática e instantaneamente após sua falta, o que torna imperceptível ao usuário e ao computador saber se está consumindo energia da rede, via fonte ou das pilhas.

Quando em teste, o INSTAN-TK foi capaz de suprir a ausência da fonte por 13 horas e 56 minutos ininterruptamente, a um consumo constante de 600 mA no modo SLOW, pois verifiquei ligeiro declínio no consumo quando o computador, em modo FAST, não mostrava a tela. Conclui-se então que o consumo varia se a tela é mostrada ou não e que o computador consome menos energia em modo FAST e rodando um programa (sem mostrar a tela) do que em SLOW ou durante uma instrução PAUSE ou ainda parado (ocasiões em que a tela é mostrada).

Condições de teste:
Computador usado: TK 82-C sem expansão de memória.
Características das pilhas: 6 pilhas novas tipo alcalina, tamanho grande, 1,5 V cada.
Circuito: INSTAN-TK original, aqui descrito.
Efeitos colaterais: nenhum efeito colateral foi observado sob condições normais de temperatura e pressão.
Melhoras adicionais: melhor imagem, livre de interferências da rede e menor aquecimento do computador.

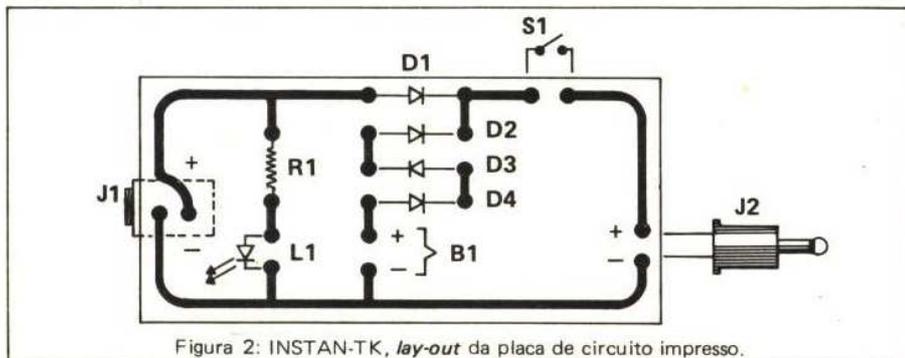



Figura 2: INSTAN-TK, *lay-out* da placa de circuito impresso.

Dependendo da aplicação específica do computador, o tempo que o aparelho é capaz de substituir a fonte pode diminuir com o uso de periféricos, tais como expansão de memória, impressora, etc. Se este for o seu caso ou se assim preferir, uma bateria de maior porte (e com consequente maior potência) de 9 Volts poderá ser usada (como aquelas usadas por fotógrafos para flashes eletrônicos, por exemplo), que ainda estará o aparelho na categoria de "temporariamente portátil", segundo os itens 1 e 2. Pode-se ainda usar pilhas recarregáveis, com a mesma vantagem da bateria de flash; longa durabilidade e possibilidade de recarga.

Por experiência própria, me sinto em posição de aconselhar a montagem e uso (ou só o uso) do INSTAN-TK pela sua grande utilidade, segurança (esqueça a tomada) e economia (de tempo e paciência). E por falar em economia (o que agrada a gregos e troianos e principalmente a árabes e judeus) o aparelho tem a característica de não inutilizar completamente as (caras) pilhas, que ainda ficam com carga suficiente para fazer um rádio portátil funcionar (principalmente as alcalinas).

A figura 2 mostra um exemplo de lay-out da placa de circuito impresso do INSTAN-TK (que, aliás, poderia muito bem ser batizado de JUMBO. Não é verdade que os elefantes jamais esquecem?) com as devidas indicações de onde e como montar os componentes. Aos pontos da placa marcados com +B1 e −B1, solde os terminais do porta-pilhas, devidamente polarizados.

Até o próximo número! ○

RESPOSTA DO
QUEBRA-CABEÇA

L.G.M.

Para nossa surpresa e grande satisfação do nosso amigo e colaborador Nabor Rosenthal, recebemos a tempo para publicar nesta edição, uma solução parcial do Quebra-Cabeça proposto no número anterior. A proeza deve-se a ***Gamal Saty***, que descobriu existir uma mensagem codificada na sequência de pulsos recebida.

Ao que nos parece, ele não teve tempo de analisar o conteúdo da mensagem, de modo que continuamos aguardando que algum de nossos leitores a interprete e traduza as informações nela contidas (essa é a segunda parte do Quebra-Cabeça).

Como todo "ovo de Colombo" a primeira parte desse Quebra-Cabeça pareceu-nos muito simples (depois de resolvida), pois é bastante lógico ao pensarmos em enviar uma mensagem ao espaço, construí-la na forma de uma matriz cujo número de elementos seja o produto de dois primos. Haja visto a mensagem enviada aqui da Terra, através do radiotelescópio de ***Arecibo***, cuja forma é a de um retângulo de 73 linhas e 23 colunas (número total de elementos $= 73 \times 23 = 1679$) – veja figura 1. Uma sequência de pulsos com essa característica (número de pulsos = produto de dois primos) é muito mais facilmente transformada em uma matriz, visto que só há uma forma de se fazer isso! No caso de nosso Quebra-Cabeça, o número total de pulsos é 931. Decompondo-o obteremos:

$$\begin{array}{r|l} 931 & 31 \\ 31 & 31 \\ 1 & \end{array} \quad \Rightarrow 931 = 31 \times 31$$

Ou seja, ele é exatamente o produto de dois primos!

Veja, a seguir, como ***Gamal*** obteve a figura a partir dos pulsos:

". . . A sequência consiste de exatamente 961 pulsos (0 e 1's) que formam um quadro de 31 linhas por 31 colunas (31 x 31).

*O primeiro programa apresentado, coloca os números diretamente numa matriz **A$(31,31)** o que facilita em muito o programa, porém o programa número dois, embora um pouco mais extenso, facilita muito a entrada de dados, pois os mesmos entram de acordo com a sequência publicada na revista, ou seja, a cada 16 pulsos, sendo assim mais fácil carregar os dados. Os mesmos são colocados numa matriz **A$ (16 x 16)**, logo em seguida reagrupados numa matriz com uma única linha e depois transformada em outra matriz (esta finalmente com 31 linhas e 31 colunas).*

O quadro formado pela sequência de 0 a 1's, indica o desenho de uma antena de micro-ondas ou mesmo um radio-telescópio, sendo possível se ver no canto superior direito, o logotipo do TK.

Este desenho foi possível, plotando-se um gráfico a partir da matriz de 31 linhas por 31 colunas.

Agora, pode-se fazer a opção:

Programa LGM1, *mais curto e simples, porém os dados entram em uma linha de 31 pulsos, apresentando certa dificuldade para se introduzir os dados da maneira como foi publicado pela revista.*

Programa LGM2, *que embora mais extenso, facilita a entrada de dados, pois o mesmo entra numa linha de 16 pulsos.*

A listagem de LGM1

```
   1 REM GAMAL SATY
   2 REM LGM1
  10 DIM A$(31,31)
  15 FOR N=1 TO 31
  20 SCROLL
  25 PRINT "A$(";N;")="
  30 SCROLL
  35 INPUT A$(N)
  40 SCROLL
  45 PRINT A$(N)
  50 SCROLL
  55 PRINT "HOUVE ERRO ? (S/N)"
  60 IF INKEY$="" THEN GOTO 60
  65 IF INKEY$="S" THEN GOTO 20
  70 NEXT N
  72 CLS
  75 REM ***********************
  80 FAST
  85 FOR I=1 TO 31
  90 FOR J=1 TO 31
  95 IF A$(I,J)="0" THEN PLOT J,
I
 100 NEXT J
 105 NEXT I
 110 STOP
 120 SAVE "LGM"
 130 PRINT "TECLE ""GOTO 80"" PA
RA EXECUTAR O PROGRAMA SEM
  PERDER OS DADOS"
 140 STOP
```

. . . Devido a certa pressa para enviar a solução, acredito que no **programa 1 (LGM1)**, *não digitei corretamente os dados, pois no* **LGM2**, *há uma pequena diferença no quadro formado que, além das características já mencionadas, apresenta também na parte inferior direita, a silhueta de um possível ser (será o E.T.?!). Como já disse, o programa 2 apresenta introdução de dados de maneira mais fácil, por isso os dados foram recolocados neste programa e que por isso, acredito estar com o desenho correto.*

Devido a certa dificuldade em chamar LGM2 regravei o programa em velocidade normal nas seguintes posições: 21Ø-247; 26Ø-3ØØ; 31Ø-355". **Gamal Saty – União da Vitória, PR.**

A listagem de LGM2

```
   1 REM GAMAL SATY
   2 REM QUEBRA-CABECA L.G.M.
  1Ø DIM A$(61,16)
  2Ø FOR N=1 TO 61
  21 SCROLL
  28 PRINT "A$(";N;")=";
  3Ø INPUT A$(N)
  35 PRINT A$(N)
  4Ø SCROLL
  41 PRINT "QUER TROCAR ALGO ?
S/N)"
  42 IF INKEY$="" THEN GOTO 42
  43 IF INKEY$="S" THEN GOTO 21
  5Ø NEXT N
  9Ø REM *********************"****
 1ØØ DIM D$(31,31)
 11Ø LET E$=A$(1)
 12Ø FOR N=2 TO 61
 13Ø LET E$=E$+A$(N)
 14Ø NEXT N
 16Ø REM ************************
 19Ø LET G=1
 192 LET H=31
 2ØØ FOR I=1 TO 31
 21Ø LET D$(I)=E$(G TO H)
 22Ø LET G=G+31
 23Ø LET H=H+31
 24Ø NEXT I
 26Ø REM **** ********************
 3ØØ FOR I=1 TO 31
 31Ø FOR J=1 TO 31
 32Ø IF D$(I,J)="Ø" THEN PLOT J,
I
 33Ø NEXT J
 34Ø NEXT I
 35Ø STOP
 36Ø SAVE "LGM█"
 37Ø PRINT "TECLE ""GOTO 3ØØ"" P
ARA EXECUTAR O PROGRAMA SEM PERD
ER OS DADOS"
 38Ø STOP
```

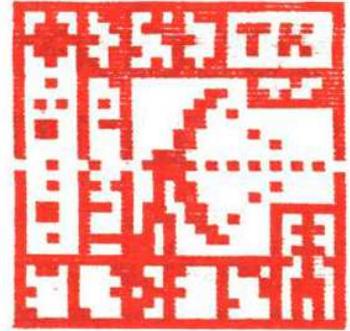



Figura 2:

Figura 1:

Números de 1 a 1Ø

Números atômicos do Hidrogênio, Carbono, Nitrogênio, Oxigênio e Fósforo.

Fórmulas dos açúcares e bases nos nucleotídeos do DNA.

Números de nucleotídeos do DNA.

Representação da estrutura do DNA.

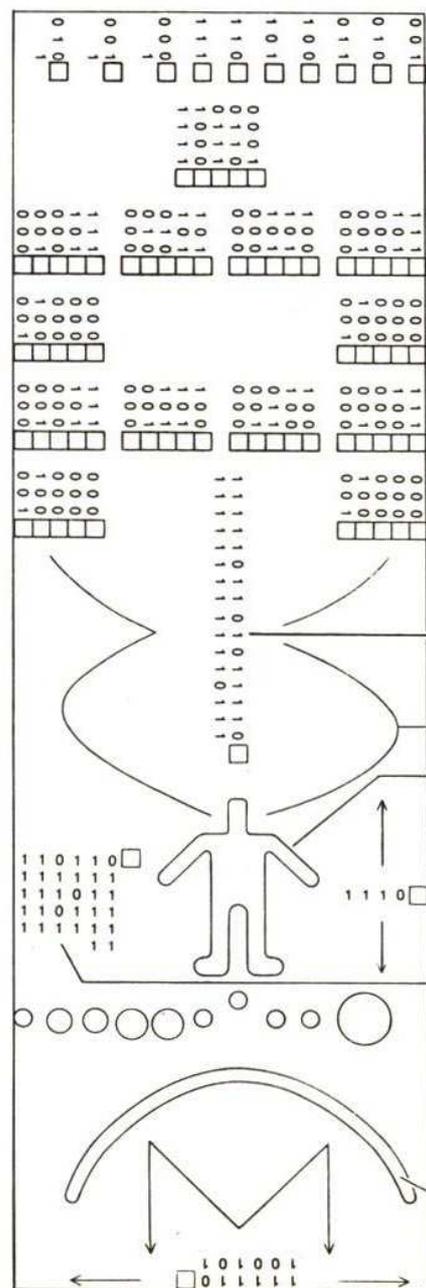
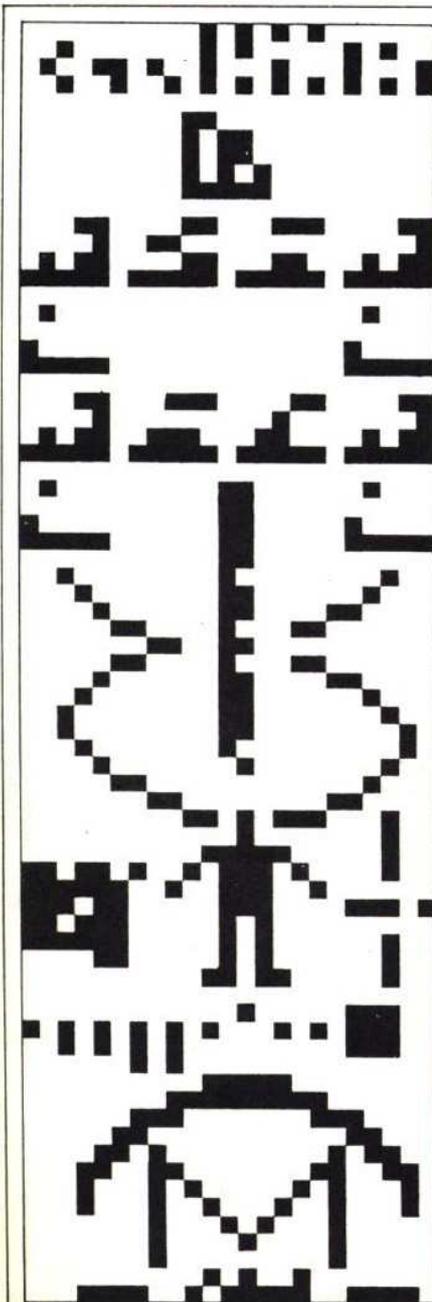
Forma humana.

Altura da forma humana.

População sobre a Terra.

Sistema solar.

Rádio-telescópio de *Arecibo.*

Diâmetro do radiotelescópio.

Mensagem enviada com o radiotelescópio de Arecibo. À esquerda, está uma figura como a obtida pelo Gamal no TK e à direita, as informações convenientemente interpretadas.

Relação das lojas autorizadas a receber assinaturas

A.D. DATA COM. SERV. DE INFORMÁTICA LTDA.
Rua João Ramalho, 818 – SP.

ALLCOLOR COM. E REPRESENTAÇÕES LTDA.
Rua Carlos Porto, 85 – Jacareí, SP
CEP 12300

BEL – BAZAR ELETRÔNICO LTDA.
Av. Almirante Barroso, 81 – Lj. C – RJ.

BRASIL TRADE CENTER COM. E PARTICIPAÇÕES S/A.
Av. Epitácio Pessoa, 280 – RJ – CEP 22471

CASA DO MICROCOMPUTADOR SIST. E PROC. DE DADOS LTDA.
Av. Anhanguera, 2574 – Centro – Goiânia, GO
CEP 74000

CENADIN – CENTRO NAC. DESENV. DA INFORMÁTICA
Rua José Maria Lisboa, 580 – SP

CESPRO – CURSOS DE ESPECIALIZAÇÃO PROFISSIONAL LTDA.
Rua República Árabe da Síria, 15 – S/207
Ilha do Governador, RJ – CEP 21931

CHIP SHOP COMPUTADORES LTDA.
Rua Ofélia, 248 – SP

COMPUTEC LTDA.
Rua Mairinque, 66 – CEP 04037 – SP.

COMPUTER CENTER MICROCOMPUTADORES MÁQ. E SISTEMAS LTDA.
Rua Lopes Trovão, 134 – Slj. 247 Center V – Icarai – Niterói, RJ – CEP 24220

COMPUTER HOUSE – JOÃO CANDIDO COLLADO
Av. Andrade Neves, 1254 – Campinas, SP

COMPUTRONIX VENDAS E SERVIÇOS LTDA.
Rua Sergipe, 1422 – Belo Horizonte, MG
CEP 30000

DATA SOLUTION LTDA.
Av. Eusébio Matoso, 654 – SP

DIDADOS INFORMÁTICA E ADM. LTDA.
Rua Minas Gerais, 655, s/602 – Divinópolis, MG

ELDORADO COMPUTADORES E SISTEMAS LTDA.
Rua Visconde de Pirajá, 351 – Lojas 213/4 – RJ – CEP 22410

ELETROSOM LTDA.
Rua da Concórdia, 287 – Recife, PE
CEP 50000

ENSICOM – ENG. DE SISTEMAS E COMP.
Rua Marques do Herval, 409 – 1º, s/15 – Taubaté, SP – CEP 12100

ENTEC REPRES. LTDA.
Rua Lauro Muller, 700 – Itajai, SC
CEP 88300

EXATRON INFORMÁTICA E ELETRÔNICA LTDA.
Alameda dos Arapanes, 841 – CEP 04524 – SP

EXPOENTE COM. E REPRESENTAÇÕES LTDA.
Av. Siqueira Campos, 838 – CEP 57000 – Maceió, AL.

GRUPO D.G.B. CONSULTORIA ADM. EMPRESARIAL S/C LTDA.
Rua Dr. Murici, 706 – ala "B", 1º – Curitiba, PR
A/C Sr. Dirceu Guimarães Brito

GUAÇUMAQ – MAQ. E EQ. P/ ESCRITÓRIO LTDA.
Rua Antonio Gonçalves Teixeira, 97 – Mogi-Guaçu, SP

GUARANI PRESENTES
Av. Senador Vergueiro, 4964 – 1º – s/6
Rudge Ramos, S.B. Campos, SP – CEP 09720

HECTOR A. FERNANDEZ – MIRAGE CINE FOTO
Rua Gal. Câmara, 648 – Sta. Barbara D'Oeste, SP – CEP 13450

LIVRARIA POLIEDRO
Rua Aurora, 704 – SP – CEP 01209

LOG COMPUTADORES LTDA.
Pça. Cândido Dias Castejon, 34 – Slj. – S. José dos Campos, SP – CEP 12200

MAPSS – ENGª COM. E IND. LTDA.
Rua Getúlio Vargas, 186 – T. Otoni, MG
CEP 39800

MAURITINO PIRES SILVEIRA
Rua Manduca Rodrigues, 924 – Sant'Ana do Livramento, RS

MEMOCARDS – MATERIAIS DIDÁTICOS LTDA.
Rua Amador Bueno, 855 – Ribeirão Preto, SP

MICROBYTE SIST. E EQUIPTOS. COMPUTACIONAIS LTDA.
Rua Buenos Aires, 41 – 3º – RJ.

MICROCENTER COMPUTADORES LTDA.
Av. Santos Dumont, 2749 – CEP 60000
Fortaleza, CE

MICRO CENTER INFORMÁTICA LTDA.
Rua Conde do Bonfim, 229 – Lj. 310/2 – Tijuca, RJ – CEP 20520

MICRODATA IMPLANTAÇÃO FÍSICA SIST. LTDA.
Rua Montreal, 16 – SP – CEP 02832

MICROESPAÇO COM. E REPRESENTAÇÕES LTDA.
Av. Barão do Rio Branco, 2288/1501 – Juiz de Fora, MG – CEP 36100

MICRON INFORMÁTICA LTDA.
Rua Benjamin Constant, 56 – sl. 804 – Viçosa, MG – CEP 36570

MICRO PROCESS COMPUTADORES LTDA.
Al. Lorena, 1310 – SP – CEP 01424

MORGEN – COM. DE COMPUTADORES
Rua Mal. Deodoro, 51 – Gal. Ritz, 14º s/1405-A – Curitiba, PR

NADAIS EQ. DE SOM LTDA.
Rua Amador Bueno, 213 – Santos, SP

NASA SHOP EQUIPTOS. ELETRÔNICOS LTDA.
Rua "4", 1042 – Centro – Goiânia, GO

NÚCLEO DE ORIENTAÇÃO DE ESTUDOS
Av. Brig. Faria Lima, 1451 – Conj. 31
01451 – S. Paulo – Fone: 813-4555

PROSERV PROC. DE DADOS CURSOS E REP. LTDA.
Lgo. 9 de Abril, 27 – s/628 – Ed. Cecisa II – Volta Redonda, RJ

RC MICROCOMPUTADORES LTDA.
Av. Estados Unidos, 983 – CEP 13400 – Piracicaba, SP.

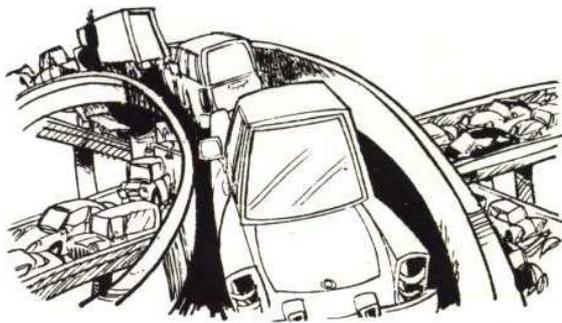
RITZ CINE FOTO LTDA.
Rua Frei Caneca, 7 – Santos, SP – CEP 11100

SHOP COMPUTER CEDM LTDA.
Av. São Paulo, 718 – Londrina, PR
CEP 86100

SIETEL SERV. INST. ELÉTRICAS E TELEC. LTDA.
Rua Cel. Joaquim Neto, 32 – Sta. Rita do Sapucaí, MG – CEP 37540

SISCOMP SISTEMAS E COMPUTADORES LTDA.
Rua Tiburcio Cavalcante, 298 – Fortaleza, CE

TELEMÁTICA COM. E IND. LTDA.
Rua Figueiredo de Magalhães, 286 – s/616
Copacabana, RJ – CEP 22031

TELEVÍDEO LTDA.
Rua Marques do Herval, 157 – Recife, PE ○

TRÂNSITO

Você se encontra de passagem em uma metrópole, com um trânsito "de doido". O seu objetivo é andar a maior quilometragem possível, na avenida principal, evitando os outros motoristas que — veja só! — insistem em seguir o caminho oposto ao seu, sem respeitar sequer a mão de direção! Além disso, você não pode subir na calçada, sob pena de perder a carta de motorista e impedido de continuar a jornada (nesta cidade, detestam forasteiros. . .)

Introdução do Programa

Este programa possui um trecho em linguagem de máquina. Para introduzí-lo, você deve inicialmente digitar o programa mostrado na figura 1. Neste programa, a linha **1 REM** deverá conter 5Ø pontos.

```
   1 REM ........................
..........................
   2 FOR I=16514 TO 16562
   3 SCROLL
   4 PRINT I;"..";
   5 INPUT N
   6 PRINT N
   7 POKE I,N
   8 NEXT I
```

fig. 1

Após a introdução, digite **RUN** e introduza os números (decimais) da listagem (figura 2).

```
LISTAGEM DECIMAL

16514..42          16515..12
16516..64          16517..17
16518..114         16519..2
16520..25          16521..229
16522..6           16523..33
16524..35          16525..15
16526..253         16527..229
16528..209         16529..225
16530..14          16531..19
16532..6           16533..33
16534..126         16535..18
16536..27          16537..43
16538..16          16539..250
16540..13          16541..32
16542..245         16543..201
16544..6           16545..24
16546..42          16547..12
16548..64          16549..35
16550..126         16551..254
16552..118         16553..40
16554..5           16555..198
16556..128         16557..119
16555..24          16559..245
16560..16          16561..243
16562..201
```

fig. 2

Ao terminar, digite **LIST**. A linha **REM** conterá os vários caracteres que correspondem aos códigos em linguagem de máquina (figura 3).

```
   1 REM E£RND)?": FAST ."57( CLE
AR FAST SGN LPRINT :(."5$4 PRINT
TAN ."/E£RND7█?/ PRINT ( NEXT TAN

   2 FOR I=16514 TO 16562
   3 SCROLL
   4 PRINT I;"..";
   5 INPUT N
   6 PRINT N
   7 POKE I,N
   8 NEXT I
```

fig. 3

Digite **POKE 1651Ø,Ø** para evitar que você distraidamente apague ou edite a linha 1, perdendo todo o programa em linguagem de máquina. Sua linha **REM** terá agora o número Ø e, desta forma, não pode ser editada, nem apagada. Uma vez tomada esta providência, apague uma por uma as linhas de dois a oito.

```
   0█REM E£RND)?": FAST ."57( CLE
AR FAST SGN LPRINT :(."5$4 PRINT
TAN ."/E£RND7█?/ PRINT ( NEXT TAN
```

A seguir, digite o programa em BASIC mostrado na figura 4.

O programa

Ao terminar de digitar o programa em BASIC, digite **RUN**. Na tela você verá as instruções. Você pode escolher cinco tipos de trânsito, digitando um número entre 1 e 5. Lembre-se: o número de motoristas doidos que aparecerão na sua frente, é diretamente proporcional ao número digitado.

Para desviar-se desses malucos, utilize a tecla 5 para mover seu carro "**$**" para esquerda e a **8** para movê-lo para a direita. No canto inferior você pode acompanhar a sua trajetória, verificando o número de quilômetros percorridos.

Caso você consiga desviar dos outros motoristas, simbolizados por "█" ou subir na calçada, o computador, um severo guarda de trânsito (mas que só pune quem é estranho na cidade, no caso você) lhe impedirá de prosseguir, tomando-lhe a carta e indicando o local do acidente.

Apesar de severo, o ***computador-guarda*** lhe dará sempre uma nova chance de recomeçar, desde que volte ao ponto inicial da avenida. Se você aceitar, digite "**S**", se não, "**N**".

Boa sorte! Mas cuidado para não ficar neurótico. ○

MEMÓRIA OCUPADA:
2.111 bytes

```
   0█REM E£RND)?": FAST ."57( CLE
AR FAST SGN LPRINT :(."5$4 PRINT
TAN ."/E£RND7█?/ PRINT ( NEXT TAN

   1 REM MICROHOBBY
   3 GOTO 2000
   4 LET HS=0
   5 RAND
   6 CLS
  10 LET X=14
  11 POKE 16418,0
  12 PRINT AT 21,0;"RECORDE:";HS
;AT 21,13;"PONTOS 0     ";AT 22,0;
"████████████████████████████████
█"
  13 POKE 16418,2
  14 PRINT AT 20,0;"███████████████
████████████████"
  20 FOR I=1 TO 19
  30 PRINT AT I,0;"000000000000
0000000000000000000";AT I,11;"██
███"
  40 NEXT I
  50 PRINT AT 0,0;"000000000000
0000000000000000000"
  60 LET L=11
  70 LET TI=0
  80 REM COMECO
 100 IF RND>.6 THEN GOTO 500
 110 LET A=USR 16514
 120 PRINT AT 1,L;" █████ "
 130 PRINT AT 11,X;"█"
 140 LET X=X+(INKEY$="8")-(INKEY
$="5")
 141 IF PEEK (PEEK 16396+256*PEE
K 16397+331+X)<>136 THEN GOTO 10
00
 145 LET TI=TI+1
 146 PRINT AT 21,13;"PONTOS:";TI
 150 PRINT AT 10,X;"$"
 170 IF RND<SK THEN PRINT AT 1,R
ND*5+L;"█"
 180 GOTO 100
 500 LET G=INT (RND*8+3)
 505 LET Q=1
 510 IF L>14 THEN LET Q=-1
 515 FOR F=1 TO G
 520 LET A=USR 16514
 525 LET L=L+Q
 530 PRINT AT 1,L;" █████ "
 535 PRINT AT 11,X;"█"
 540 LET X=X+(INKEY$="8")-(INKEY
$="5")
 552 IF PEEK (PEEK 16396+256*PEE
K 16397+331+X)<>136 THEN GOTO 10
00
 555 LET TI=TI+1
 557 PRINT AT 21,13;"PONTOS:";TI
 560 PRINT AT 10,X;"$"
 580 IF RND<SK THEN PRINT AT 1,R
ND*5+L;"█"
 590 NEXT F
 600 GOTO 100
1000 PRINT AT 10,X-1;"" /.";AT 11
,X-1;">*""
1005 FOR F=1 TO 26
1006 LET C=USR 16544
1007 NEXT F
1010 POKE 16418,0
1015 IF TI>HS THEN LET HS=TI
1020 PRINT AT 21,0;"RECORDE:";HS
;AT 23,8;"OUTRO JOGO (S/N)?"
1030 POKE 16418,2
1040 IF INKEY$="S" THEN GOTO 6
1050 IF INKEY$="N" THEN STOP
1060 GOTO 1040
2000 REM INSTRUCOES
2020 PRINT AT 0,0;"************ T
RANSITO ************"
2030 PRINT AT 2,0;"NESTE JOGO AS
 PRINCIPAIS","INSTRUCOES SAO:"
2032 PRINT
2033 PRINT
2034 PRINT
2040 PRINT ,,,,"5 (MOVER P/ ESQU
ERDA)","8 (MOVER P/ DIREITA)"
2050 PRINT "SEU CARRO E O ""$"""
2060 PRINT "VOCE DEVE EVITAR OS
OUTROS","CARROS, QUANDO ELES VIE
REM","AO SEU ENCONTRO DO TIPO ""
█"""
2070 PRINT
2080 PRINT "ESCOLHA O TRANSITO (
1 A 5)"
2090 INPUT SK
2100 IF SK<1 OR SK>5 THEN GOTO 2
090
2110 LET SK=SK/10
2120 GOTO 4
2130 STOP
3000 SAVE "TRANSITO"
3010 RUN
3020 STOP
```

fig. 4

Muitos de nós provavelmente já tiveram o desejo, ou talvez apenas a idéia, de alterar, de alguma forma, a linguagem BASIC, para obter algum efeito especial no seu computador. O maior problema reside no fato de que, estando a linguagem gravada em ROM (e não em RAM, como é feito em alguns computadores), não é possível alterar o seu conteúdo através de comandos no teclado.

O primeiro impulso dos mais interessados seria o de fazer, ou mandar fazer, um novo conjunto de ROMs, com as modificações desejadas, e substituir os originais. Porém, uma pessoa mais criativa, que conheça um pouco do ***hardware*** do Apple, logo lembrará que existe o cartão periférico de 16K, que pode, sob controle do computador, substituir a ROM principal. Todo o seu conteúdo pode ser transferido à RAM do cartão, e as modificações efetuadas por **POKE**s ou pelo monitor e gravadas num disco.

A nossa equipe, pensando nisso, decidiu tomar esse caminho, desenvolvendo um pequeno programa que serve como exemplo para os que quiserem fazer as suas modificações no BASIC. O programa permite ao seu usuário alterar as palavras reservadas do **Applesoft,** podendo substituir, por exemplo, **HOME** por **CLS** ou **PRINT** por **MOSTRE**. O programa, como o leitor já deve ter percebido, exige a presença do chamado ***language card,*** o cartão de expansão de 16K, sem o que este não funcionará como esperado. Para os que não tiverem este periférico, veremos, nos próximos artigos, como alterar este mesmo programa para permitir a modificação dos comandos de DOS, que reside em RAM.

O programa

Antes de podermos entrar no funcionamento do programa, devemos conhecer o formato em que estão armazenadas as palavras reservadas no **Applesoft**. A lista de 107 palavras se encontra gravada, seqüencialmente, a partir da posição 53456. Elas estão codificadas em ASCII, com o bit 7 no estado 1 indicando o final de uma palavra. Como as primeiras palavras são **END** e **FOR**, os primeiros 6 bytes são:

posição	conteúdo	ASCII
53456	69	E
53457	78	N
53458	196	D
53459	70	F
53460	79	O
53461	210	R

O programa começa colocando numa matriz as palavras reservadas, separando-as quando o código de um caractere é maior que 127 (bit 7 = 1), estando definidas, nas variáveis P e NP, respectivamente, a posição de memória inicial da lista de palavras e o número de palavras na lista (linhas 100-140). Como a lista tem tamanho fixo na memória, não podemos aumentar seu tamanho total, que é a soma dos tamanhos das palavras individuais. Para que esse limite não seja ultrapassado, a variável NB mantém a contagem do número de bytes que sobram depois de uma alteração.

A seguir, o programa pede uma palavra, coloca um 1 no bit 7 da última letra (linhas 150-170) e procura palavra na matriz A$( ). Não existindo, é dado um aviso; se a palavra digitada existe na lista, é pedida uma nova palavra para substituí-la, com o cuidado de não ultrapassar o tamanho máximo da lista (linhas 180-200). A contagem de bytes sobrando é atualizada, o último byte da nova palavra tem seu bit 7 igualado a 1, somando-lhe 128, a palavra é substituída na matriz, e a execução passa à linha 150 para pedir uma nova palavra ou terminar (linhas 210-220).

No final, o usuário tem a opção de fazer ou não, a alteração final no cartão de RAM (230). Se ele quiser fazê-lo, o ***language card*** é preparado para receber a cópia da ROM e é instalada na memória, um pequeno programa em linguagem de máquina para chamar a rotina **MOVE** do monitor, junto com os parâmetros necessários para a sua execução (240). Partindo da posição P, a linha 250 **POKE**eia no cartão de expansão, caractere por caractere, o conteúdo da matriz de palavras A$( ). O **POKE** da linha 260 faz com que o RAM do cartão substitua o ROM da memória principal, tornando efetivas as modificações feitas (até que, é claro, seja desligado o computador).

A linguagem modificada pode ser armazenada em disco com um

```
BSAVE nome,A$D000,L$2FFF
```

Depois, para reinicializar o cartão, execute os seguintes comandos:

```
POKE -16255,0: POKE -16255,0
BLOAD nome
POKE - 16256,0
```

É claro que, se você quiser incluir esses comandos num programa, o segundo comando deve ser substituído por:

```
PRINT CHR$(4); "BLOAD nome"
```

Algumas observações

1. Quaisquer programas escritos anteriormente à uma modificação deste tipo na linguagem, não precisam (nem devem) ser alterados, pois o BASIC não armazena a palavra reservada e sim um código que a representa no programa.

2. Este programa não deve ser utilizado num Applesoft já modificado, a não ser que se tenha completo controle sobre o que está acontecendo no computador.

3. Lembre-se de que, se houver algum problema durante ou depois da execução do programa de modificação, o simples acionamento do interruptor do computador o resolverá.

4. Com algumas modificações, este mesmo programa pode ser usado para modificar as palavras do DOS, sem a necessidade de se ter um cartão de expansão de memória. Nos próximos números mostraremos como fazê-lo.

5. Alguns leitores já devem ter pensado em modificar as mensagens de erro. Utilizando uma técnica semelhante (mas não igual, devido a uma diferença que há no armazenamento dessas mensagens) veremos, nos próximos números, como fazer isso no Applesoft e no DOS.

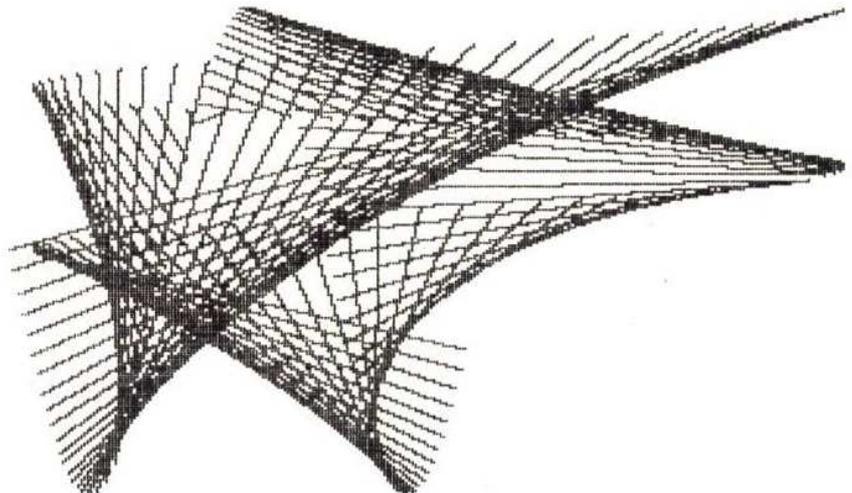
6. Comentários dos leitores a respeito deste programa, ou de outros relacionados ao assunto, serão bem aceitos pela nossa equipe.

```
0  REM ** TRADUTOR DE PALAVRAS **

   @1983 POR DENTRO DO APPLE
100 P = 53456:NP = 107
110  DIM A$(NP):NB = 0:J = P - 1:
      HOME : PRINT  TAB( 4);"TRAD
     UTOR DE PALAVRAS RESERVADAS"
     : PRINT : PRINT : PRINT "AGU
     ARDE..."
120  FOR I = 1 TO NP:A$(I) = ""
130 J = J + 1:A$(I) = A$(I) +  CHR$
     ( PEEK (J)): IF  PEEK (J) <
     128 THEN 130
140  NEXT I
150  PRINT : PRINT : PRINT "SOBRA
     M ";NB;" BYTES.": PRINT "DIG
     ITE O COMANDO A ALTERAR --":
      PRINT " <RETURN> PARA TERMI
     NAR:": INPUT "";A$: IF A$ =
     "" THEN 230
200 NL =  LEN (A$(I)) + NB: PRINT
     : PRINT "SUBSTITUI'-LO POR (
     ATE' ";NL;" CARACTERES):": INPUT
     "";B$:L =  LEN (B$): IF L >
     NL OR L = 0 THEN  PRINT  CHR$
     (7);: GOTO 200
210 NB = NB +  LEN (A$(I)) - L: IF
     L = 1 THEN A$(I) =  CHR$ ( ASC
     (B$) + 128): GOTO 150
220 A$(I) =  LEFT$ (B$,L - 1) +  CHR$
160  IF  LEN (A$) = 1 THEN A$ =  CHR$
     ( ASC (A$) + 128): GOTO 180
170 A$ =  LEFT$ (A$, LEN (A$) - 1
     ) +  CHR$ ( ASC ( RIGHT$ (A$
     ,1)) + 128)
180  FOR I = 1 TO NP: IF A$(I) =
     A$ THEN 200
190  NEXT I: PRINT : PRINT "ESSA
     PALAVRA NAO EXISTE"; CHR$ (7
     ): GOTO 150
     ( ASC ( RIGHT$ (B$,1)) + 128
     ): GOTO 150
230  INPUT "POSSO FAZER A ALTERAC
     AO FINAL  (S/N)? ";A$: IF  LEFT$
     (A$,1) = "N" THEN  HOME : PRINT
     "NENHUMA ALTERACAO FOI FEITA
      - TERMINADO.": END
240  FOR I = 1 TO 13: READ X,Y: POKE
     X,Y: NEXT I: CALL 768
250  PRINT : PRINT "AGUARDE...": FOR
     I = 1 TO NP: FOR J = 1 TO  LEN
     (A$(I)): POKE P, ASC ( MID$
     (A$(I),J,1)):P = P + 1: NEXT
     J,I
260  POKE 49280,0: HOME : PRINT "
     FIM DA ALTERACAO.": END
270  DATA  768,160, 769,0, 770,76
     , 771,44, 772,254, 49281,0,
     49281,0, 60,0, 61,208, 62,25
     5, 63,255, 66,0, 67,208
```

TWO LINERS

Tio Fred List agradece a todos os que têm enviado programas e em especial a Guilherme Nogueira Rodrigues o **two liners** enviado, cuja listagem vem a seguir. A **coisa** que esse programa cria está mostrada na figura em anexo.

```
0  REM GUILHERME N. RODRIGUES
        SAO PAULO - SP
1  HGR2 : HCOLOR= 3: FOR X = 0 TO
     190:P = X *  ATN (1) / 45 *
     360 / 279 * 4:X2 = X2 + 279 /
     191:P2 = X2 *  ATN (1) / 45 *
     360 / 279 * 3
2 Y1 =  SIN (P) * 95 + 95:Y2 =  COS
     (P2) * 130 + 139: HPLOT X2,Y
     1 TO Y2,X: NEXT
```

Enviem suas colaborações para a "Seção Por Dentro do Apple" no seguinte endereço:
MICROMEGA P.M.D. LTDA.
Rua Bahia, 1049
01296 – São Paulo – S.P.

Colaborações de Daniel Falconer e José Eduardo Moreira professores assistentes do Departamento de Computação da Escola Pueri Domus.

TIG-COMP

Gustavo Egídio de Almeida

Tig-Comp é um típico programa-ferramenta, cuja função é compilar os programas em Basic, ou seja, transformar os programas feitos originalmente em Basic, em programas de linguagem de máquina do Z8Ø, linguagem essa muito mais veloz que o Basic.

Este software é comercializado pela Tigre-Comércio de Equipamentos para Computadores Ltda.

O programa nos foi enviado, gravado numa fita de boa qualidade, acompanhada de manual explicativo. (Fig. 1)

fig. 1

Tig-Comp pode ser usado em todos os computadores compatíveis com a linha Sinclair, TK-82, TK-83, TK-85, CP-200, e outros.

Testando **Tig-Comp** em alguns programas desenvolvidos ou adaptados na nossa redação, este veio a confirmar as expectativas, provando ter um bom desempenho apesar de haver algumas restrições, mas quase todas superáveis.

A fita trabalha somente com números inteiros, na faixa de –32768 a 32767, sem operações com "Strings" e Álgebra Booleana.

Os comandos usados pelo **Tig-Comp**, são os seguintes:

ABS, CHR$, CLS, FAST, GOSUB, GOTO, SCROLL, SLOW, STOP, UNPLOT, USR e **COPY** (usando-se USR 2153).

DIM Z(V) – A matriz deve ser única, devendo ser dimensionado apenas um ARRAY sempre tendo o nome Z.

FOR-NEXT – Seu incremento é sempre de 1.

Exemplo:

Maneira errada:

```
10 FOR C=1 TO 10 STEP 2
20 NEXT C
```

Maneira correta:

```
10 FOR C=1 TO 10
15 LET C=C+1
20 NEXT C
```

IF-THEN –

Exemplo:

Maneira errada:

```
10 IF A=1 OR B=2 THEN GOTO 100
```

Maneira correta:

```
10 IF A=1 THEN GOTO 100
20 IF B=2 THEN GOTO 100
```

LET: Este comando é restrito às quatro operações (+, –, *, /) mais as seguintes funções:

LET V = (PEEK U, USR U, ABS U, INT U)

LET V = RND: Esta função é diferente da usada no seu micro. No TK, o RND gera um número aleatório de Ø a 1. No **Tig-Comp,** RND gera um número inteiro de Ø a 32767. Para compreender melhor o uso desta função, compare os exemplos:

Maneira errada:

```
10 LET L=INT (RND*30)
```

Maneira correta:

```
10 LET L=INT (RND/(32767/30))
OU
10 LET L=INT (RND/1092)
```

Note que, nesta função é gerado um número de Ø a 32767, que deve ser dividido pelo número que faríamos variar, se usássemos essa mesma função no Basic-TK:

LET V = CODE INKEY$ – Assume o valor do código da tecla pressionada.

Exemplo:

Maneira errada:

```
10 IF INKEY$="5" THEN LET A=A+1
```

Maneira correta:

```
10 LET J=CODE INKEY$
20 IF J=33 THEN LET A=A+1
```

NEW – Usa-se USR Ø se desejar "apagar" toda a memória, incluindo acima da RAMTOP.

Usa-se USR 963 para "apagar" até a RAMTOP, como usado normalmente.

PAUSE V – A tela não oscila. O valor de **V** tem que ser menor que 32768 e qualquer tecla pressionada durante a pausa, provocará a execução do comando seguinte:

BREAK – Retorna ao Basic.

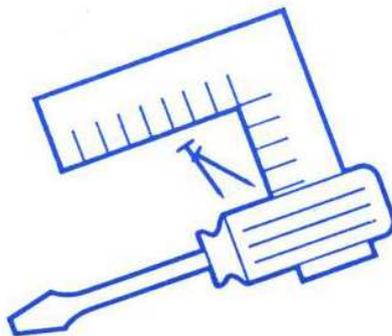
PRINT – Podem ser usados: PRINT AT, "STRING", VARIÁVEL, CHR$, podendo os mesmos serem combinados.

Exemplo:

```
10 PRINT AT 10,10;"MICROMEGA";AT
 11,10;"MICROHOBBY"
```

Agora apresentaremos um ***programa-exemplo***, mostrando-o primeiramente em Basic-TK e, posteriormente, o mesmo programa já com as respectivas modificações, podendo ser compilado pelo **Tig-Comp.**

Antes de tudo, devemos carregar o **Tig-Comp** com o comando LOAD " ". Carregado o programa, devemos deletar todas as linhas menos as de números Ø, 2, 3 e 4; que serão usadas pelo compilador.

Seu programa poderá então ser digitado a partir da linha 5.Vamos ao ***programa exemplo:*** (Figura 2)

Agora, atenção: na figura 3 mostramos esse mesmo programa, modificado apenas para sua aceitação no **Tig-Comp.**

```
   5 REM METEORO
  10 REM GUSTAVO EGIDIO DE
         ALMEIDA
  15 LET B=10
  20 LET A=10
  25 LET C=0
  30 IF INKEY$="5" THEN LET B=B-
1
  35 IF INKEY$="8" THEN LET B=B+
1
  37 IF B<0 THEN LET B=0
  38 IF B>29 THEN LET B=29
  40 PRINT AT A,B;"■";AT A,B;" "
  45 LET C=C+1
  50 PRINT AT A+1,B;
  55 IF PEEK (PEEK 16398+256*PEE
K 16399)<>0 THEN GOTO 75
  60 SCROLL
  65 PRINT AT 20,INT (RND*30);"*
";AT 20,INT (RND*3C);"*";AT 20,I
NT (RND*30);"*"
  70 GOTO 30
  75 PRINT AT 20,7;"PONTOS :";C
  80 PRINT AT 21,0;"QUER JOGAR D
E NOVO ? (S/N)"
  85 IF INKEY$="S" THEN GOTO 100
  90 IF INKEY$="N" THEN PRINT AT
 10,13;"■■■"
  95 GOTO 85
 100 CLS
 105 GOTO 15
```

fig. 2

```
   5 REM METEORO
  10 REM GUSTAVO EGIDIO DE
         ALMEIDA
  15 LET B=10
  20 LET A=10
  25 LET C=0
  30 LET J=CODE INKEY$
  32 IF J=33 THEN LET B=B-1
  35 IF J=36 THEN LET B=B+1
  37 IF B<0 THEN LET B=0
  38 IF B>29 THEN LET B=29
  40 PRINT AT A,B;"■";AT A,B;" "
  45 LET C=C+1
  50 PRINT AT A+1,B;
  55 IF PEEK (PEEK 16398+256*PEE
K 16399)<>0 THEN GOTO 75
  60 SCROLL
  65 PRINT AT 20,INT (RND/1092);
"*";AT 20,INT (RND/1092);"*";AT
20,INT (RND/1092);"*"
  70 GOTO 30
  75 PRINT AT 20,7;"PONTOS :";C
  80 PRINT AT 21,0;"QUER JOGAR D
E NOVO ? (S/N)"
  83 LET J=CODE INKEY$
  85 IF J=56 THEN GOTO 100
  90 IF J=51 THEN PRINT AT 10,13
;"■■■"
  95 GOTO 83
 100 CLS
 105 GOTO 15
 110 STOP
```

fig. 3

Especial atenção às linhas 3Ø, 32, 35 e 65. (Figura 3)

Após digitar-se o novo programa, teremos que compilá-lo, ou seja, transformá-lo de Basic para linguagem de máquina.

Para isso temos que acionar o **Tig-Comp** através do comando direto, ou seja, sem número de linha:

LET L = USR 17389

A operação de compilar é rápida, e após o comando acima, sua listagem aparecerá na tela em forma de SCROLL.

Essa operação se repetirá três vezes e logo após isso, aparecerá no canto inferior da tela o código Ø/Ø, finalizando a operação do compilador. A partir deste momento, seu programa já estará compilado.

Se seu programa foi verificado, atentamente, em Basic, poucos erros surgirão. Caso isso ocorra, surgirá um **S** inverso, sobre ou próximo ao erro cometido.

O cursor é transposto instantaneamente à linha não identificada pelo compilador, faltando somente editar e corrigí-la.

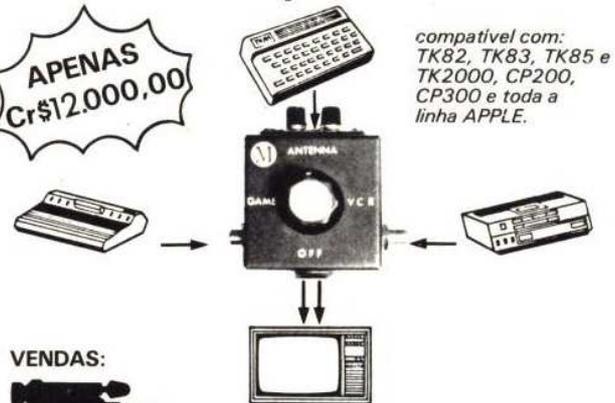
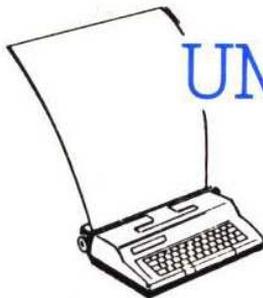
Depois de compilado, o programa deve ser tratado como uma rotina em linguagem de máquina e, para acioná-lo, usa-se o comando LET L = USR 18823. ○

ERRATA

O ESQUIADOR, MICROHOBBY Nº 6

A NEVE DO ESQUIADOR REALMENTE ERA "FOFA" E DESLIZOU SOBRE NÓS. A "AVALANCHE" SE DEVE AO FATO DE TERMOS OMITIDO OS CÓDIGOS DE LINGUAGEM DE MÁQUINA DA LINHA REM.BASICAMENTE NÃO HÁ PROBLEMA EM COPIA-LA CARACTERE POR CARACTERE(VEJA "DESGRILANDO" DO Nº 5). PORÉM, UM DOS CARACTERES ESTÁ EM INVERSO E ALGUMAS REVISTAS PODEM TER SAÍDO COM UM PEQUENO DEFEITO NA IMPRESSÃO, IMPEDINDO SUA LEITURA.

ASSIM OS CÓDIGOS SÃO:

```
16514...42     16515...12
16516...64     16517...229
16518...17     16519...33
16520...0      16521...25
16522...209    16523...1
16524...214    16525...2
16526...237    16527...176
16528...201
```

UM EDITOR DE CARTAS

Última Parte

Igor Sartori

As funções de "fatiamento" de ***strings***, também conhecidas por funções ***slicings***, são convenientes para qualquer programa de processamento de textos.

Estas funções permitem ao computador "quebrar" uma cadeia de caracteres em pontos determinados. Estas funções, associadas com as outras que descrevemos anteriormente (**Microhobby** número 6), nos permite isolar qualquer grupo de caracteres numa *string* e executar funções com eles.

RIGHT$, MID$ e LEFT$

As funções **RIGHT$, MID$** e **LEFT$** são responsáveis no TRS-80 e na maioria dos computadores (mas não no TK) pela *slicing.*

RIGHT$ *(string ou variável string,* **n***)*: esta função fornece os últimos caracteres da *string*, ou seja, os **n** caracteres à direita, contados a partir do último.

Por exemplo, no programa:

```
10 LET A$ = "CERTAMENTE"
20 LET B$ = RIGHT$ (A$,5)
30 PRINT B$
```

O computador exibirá na tela:

MENTE

LEFT$ *(string ou variável string,* **n***)*: fornece os **n** primeiros caracteres de uma *string.*

Por exemplo, no programa:

```
10 LET A$ = "CERTAMENTE"
20 LET B$ = LEFT$ (A$, 5)
30 PRINT B$
```

O computador exibirá na tela:

CERTA

MID$ *(string, ou variável string* **n**, **m***)*: fornece um trecho da *string* de comprimento **n** a partir do caractere **m**. Por exemplo, o programa:

```
10 LET A$ = "CERTAMENTE"
20 LET B$ = MID$ (A$,3,4)
30 PRINT B$
```

exibirá na tela:

TAM

Além destas funções, podemos juntar pedaços de ***strings*** ou ***strings*** inteiras por meio da operação de concatenação ou soma de ***strings.*** Esta operação "soma" duas ***strings*** e fornece como resultado uma ***string*** com os caracteres justapostos. Por exemplo, o programa:

```
10 LET A$ = "CERTA"
20 LET B$ = "MENTE"
30 LET C$ = A$ + B$
40 PRINT C$
```

exibirá na tela:

CERTAMENTE

O slicing no TK-83

No TK-83 e seus compatíveis é possível fazer o "fatiamento" de *strings* usando-se um recurso muito interessante: a própria posição dos caracteres.

Se você conhece bem o TK, deve estar lembrado que cada caractere ocupa uma posição de memória numa variável *string.* Cada uma destas posições de memória tem o tamanho exato de um byte. O BASIC do TK permite que tenhamos acesso à cada um destes bytes. Por exemplo na figura 1, mostramos um programa em BASIC TK que nos permite acessar um a um os bytes da *string* "CERTAMENTE".

```
10 LET A$="CERTAMENTE"
20 FOR I=1 TO 10
30 PRINT TAB (12);A$(I)
40 NEXT I

            C
            E
            R
            T
            A
            M
            E
            N
            T
            E
```

Com este tipo de artifício poderemos simular no TK, as instruções RIGHT$, MID$ e LEFT$.

RIGHT$: no TK, devemos usar artifício que faça o computador considerar a ***string*** de um ponto qualquer até o final. Para isso, basta fazer:

```
LET B$=A$(N TO )
```

onde **A$** e **B$** são variáveis *strings,* e **N** é um número inteiro. Repare que após o **TO** (SHIFT 4) não aparece número algum. Assim, o computador considera a *string* a partir do caractere **N** até o final.

Experimente agora o programa da figura 2. Ele fornece todas as partes de uma *string,* que podem ser obtidas por este artifício.

```
10 LET A$="CERTAMENTE"
20 FOR I=1 TO 10
30 PRINT A$(I TO )
40 NEXT I

CERTAMENTE
ERTAMENTE
RTAMENTE
TAMENTE
AMENTE
MENTE
ENTE
NTE
TE
E
```

LEFT$: basta fazer com que o TK considere uma ***string*** a partir de primeiro até o "enésimo" caractere. Por exemplo:

```
LET B$=A$(1 TO N)
```

onde **N** significa o número de caracteres que desejamos na ***string.*** O programa da figura 3 é um bom exemplo de como as coisas ocorrem no TK.

```
10 LET A$="CERTAMENTE"
20 FOR I=1 TO 10
30 PRINT A$(1 TO I)
40 NEXT I

C
CE
CER
CERT
CERTA
CERTAM
CERTAME
CERTAMEN
CERTAMENT
CERTAMENTE
```

MID$: basta fazer com que o TK considere uma ***string*** a partir de um caractere também determinado. Por exemplo:

```
LET B$=A$(N TO M)
```

onde **N** representa o primeiro caractere considerado e **M** o último. O programa da figura 4 mostra um exemplo de aplicação para este tipo de instrução.

```
10 LET A$="CERTAMENTE"
20 FOR I=2 TO 9
30 PRINT A$(I-1 TO I+1)
40 NEXT I

CER
ERT
RTA
TAM
AME
MEN
ENT
NTE
```

Tanto no TK como no TRS-80 estas funções podem ser associadas a outras para se calcular a posição dos caracteres de uma *string*.

Implementando o Editor de Cartas

Com o conhecimento que já temos, podemos traduzir o programa do TRS 80 para o TK. Na figura 5, mostramos uma listagem de como o programa ficou após nossas traduções. Evidentemente, cortamos uma parte dele para simplificá-lo. Todavia, as alterações que fizemos não foram muito profundas e podem ser incrementadas na listagem final.

```
   1 REM EDITOR
  10 LET E=0
  30 DIM A$(20,32)
  40 PRINT "TAMANHO MAX. DA LINH
A=32 LETRAS"
 130 PRINT "INSTRUCOES"
 140 PRINT "INTRODUZA AS LINHAS
1 A 1"
 150 PRINT "P/ CORRIGIR 1 LINHA"
 160 PRINT "DIGITE"">""E O SEU N
• "
 170 PRINT "DIGITE FIM PARA PARA
R"
 171 PRINT "QDO A TELA ESTIVER C
HEIA"
 173 PRINT "DIGITE CONT"
 175 PAUSE 300
 176 CLS
 180 FOR X=1 TO 100
 190 LET Z=X
 200 PRINT ">";X
 210 INPUT A$(X)
 215 PRINT A$(X)
 220 IF A$(X,1)=">" THEN GOTO 45
0
 230 IF A$(X,1 TO 3)="FIM" THEN
GOTO 260
 250 NEXT X
 260 CLS
 340 FOR B=1 TO X-1
 350 PRINT A$(B)
 440 STOP
 450 REM ROTINA DE CORRECAO
 460 LET Y=VAL A$(X,2)
 470 PRINT "SELECAO"
 471 PRINT "0-INTRODUZIR"
 472 PRINT "1-CORRECAO"
 475 INPUT A
 476 CLS
 480 IF A<0 OR A>1 THEN GOTO 470
 490 GOTO 500+A*20
 500 LET A$(Y)=""
 505 INPUT A$(Y)
 506 PRINT A$(Y)
 510 LET X=X-1
 511 GOTO 250
 520 LET T=LEN A$(Y)
 530 LET C=1
 540 LET M=0
 550 LET Q$=INKEY$
 560 IF Q$="0" THEN GOSUB 1000
 570 IF Q$="D" AND C<=T THEN GOS
UB 2000
 580 IF Q$="I" THEN GOSUB 3000
 590 IF Q$="E" THEN GOSUB 4000
 600 IF Q$="L" THEN GOSUB 5000
 620 GOTO 550
 900 STOP
1000 REM MOSTRA CARACTERES
1005 IF C>T THEN GOTO 550
1010 LET M=M+1
1020 LET C=C+1
1040 LET I$=A$(Y,M)
1050 PRINT I$;
1060 RETURN
2000 REM DELETA CACTERE
2010 LET A$(Y)=A$(Y,1 TO C-1)+A$
(Y,C+1 TO )
2020 RETURN
3000 REM INCLUSAO
3010 INPUT J$
3020 LET A$(Y)=A$(Y,1 TO C-1)+J$
+A$(Y,C TO (32-C-LEN J$))
3030 RETURN
4000 REM ESCAPE
4020 PRINT
4030 PRINT A$(Y)
4040 LET A$(X)=""
4050 LET X=X-1
4060 GOTO 240
5000 REM MOSTRA LINHA
5010 PRINT A$(Y,C TO )
5020 LET M=0
5030 LET C=1
5040 RETURN
```

Vamos, agora, descrever algumas das alterações que foram necessários para que o programa "rodasse" no TK.

A primeira alteração importante está na linha 3Ø. Nela ao invés de fazermos:

```
DIM A$(100)
```

fizemos:

```
DIM A$(20,32)
```

Isso porque limitamos o número de linhas à 20, com 32 caracteres. No TK, o último índice de uma matriz *string* é o indicativo do tamanho máximo que cada variável, que faz parte desta matriz, pode ter. Assim, no nosso caso, A$(20,32) significa que temos uma matriz com 20 variáveis, contendo 32 caracteres cada uma. O segundo índice é usado no *slicing* indicando ao computador que aquela variável possui um comprimento máximo (no nosso caso, 32). Cuidado entretanto com as "atribuições procustianas".

As Atribuições Procustianas

Uma das coisas que mais tem "grilado" os possuidores do TK-83 e seus compatíveis é que, nos manuais de muitas destas máquinas, aparece uma frase com mais ou menos este conteúdo: "As atribuições de uma variável *string* só podem ser procustianas".

Isso apareceu no manual, porque seu autor conhecia muito bem Mitologia: *Procustes* era um gigante que raptava as pessoas e as colocava deitadas numa cama. Se a pessoa fosse menor, ele as esticava até que ocupassem todo o leito. Se fossem maiores ele cortava suas pernas no comprimento da cama. É o que acontece com a atribuição de variáveis no TK. Quando você dimensiona um variável com **DIM A$**(10), ao atribuirmos uma *string* à **A$**, o computador sempre considera a *string* com dez caracteres (o valor do índice): se a *string* for maior, ele corta os últimos caracteres; se for menor, ele preenche o que falta para completar, com espaços em branco. Experimente rodar o programa que mostramos na figura 6 para entender melhor o que ocorre.

```
10 DIM A$(10)
20 LET A$="PARALELEPIPEDO"
30 PRINT A$,LEN A$
40 LET A$="PARALELO"
50 PRINT A$,LEN A$


PARALELEPI      10
PARALELO        10
```

Como você deve ter notado, as atribuições deste tipo fazem com que as *strings* tenham sempre o mesmo comprimento. Levando isso em consideração não poderemos, em hipótese alguma, controlar estas variáveis pelo seu comprimento, como fizemos no TRS-80.

Outras modificações

Algumas modificações são óbvias e não as comentaremos (basta que você dê uma olhada na listagem que apresentamos neste número e na listagem que apresentamos na **Microhobby** 5).

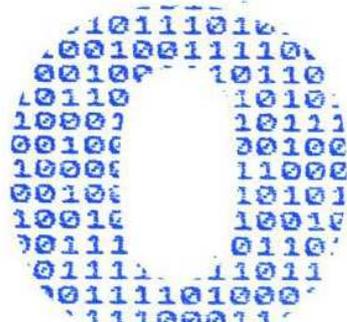
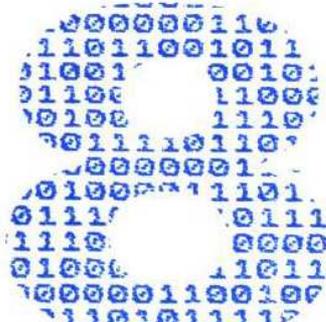
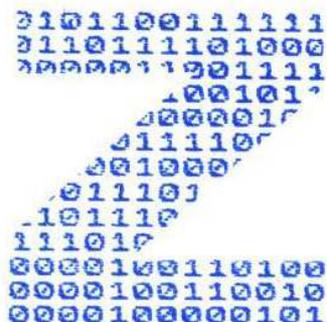
Devido às "atribuições procustianas", eliminaremos a parte que verifica se uma linha digitada excedeu o tamanho máximo. Desta forma, se uma linha for excedida, o computador desconsidera o trecho a mais.

As rotinas de correção foram substituídas por sub-rotinas. Atenção para a sub-rotina que mostra os caracteres, uma vez que o BREAK do TK está na mesma tecla que o espaço, usamos o número Ø para fazer esta função.

Com o que descrevemos neste e nos números anteriores, com um pouco de atenção, você pode analisar uma a uma as rotinas de correção de erros. Note que não fizemos uma rotina para o *backspace* (retorno do cursor sem apagamento de caracteres). Para isso, devemos lançar mão de uma rotina em linguagem de máquina. Isso fica como uma sugestão para que você desenvolva uma e nos envie como colaboração (se for em BASIC, melhor ainda).

Como qualquer outra seção desta revista, a seção* Os Oitenta *está aberta às colaborações, desde que estejam dentro do seu espírito: artigos voltados para o TRS-80, de preferência sem mencionar-se discos ou linguagem de máquina, muito bem explicados, com detalhes técnicos para que o usuário do TK possa aprender um pouco mais de programação e conhecer uma outra máquina. Daremos preferência a artigos e programas que procurem aproximar as duas máquinas, mostrando suas semelhanças e diferenças.

O

CURSO DE ASSEMBLY
aula 6

Flavio Rossini

Iremos agora fazer "contas" com os registradores, assim como fazemos em BASIC, com as variáveis de um programa; entretanto, a linguagem de máquina permite fazer diretamente apenas contas de soma e subtração e, como você já deve ter notado, lida apenas com números inteiros.

Vamos indicar com as instruções de adição de registros e de pares de registros: (Tabela 1) e (Tabela 2)

INSTRUÇÕES		CÓDIGO
ADD	A,A	'87'
ADD	A,B	'80'
ADD	A,C	'81'
ADD	A,D	'82'
ADD	A,E	'83'
ADD	A,H	'84'
ADD	A,L	'85'

Tab. 1

INSTRUÇÕES		CÓDIGO
ADD	HL,BC	'09'
ADD	HL,DE	'19'
ADD	HL,HL	'29'

Tab. 2

A instrução ADD entre registros.

que significam: some o conteúdo do registro ou par de registros da direita com o da esquerda e conserve o resultado no registro, ou par de registros da ESQUERDA; os registros da direita, portanto, NÃO são alterados. Em BASIC, uma analogia poderia ser feita com:

LET X = X + Y

Novamente notamos o registro A, ou seja, o ACUMULADOR como sendo ***privilegiado***; de fato, ele é o "alvo" ***de todas as operações aritméticas*** entre registradores ou entre registradores e memória, permanecendo nele o resultado da operação. No caso de par de registros, o ***par privilegiado*** é o ***HL***. Assim, qualquer operação aritmética DEVE utilizar o acumulador ou o par ***HL***.

Vamos então tentar somar o conteúdo do registro ***D*** com o registro ***E*** com uma subrotina em linguagem de máquina: (Tabela 3).

Note que a soma ***não pode ser feita diretamente***! Devemos obrigatoriamente usar o ACUMULADOR. Coloque então o programa na memória e execute-o (usando HEXAMEM, publicado na edição número 4 e na número X, que condensa os melhores programas publicados na MICROHOBBY). Voce deverá obter o número 100 (= 34 + 66)! Provavelmente deve ter surgido na sua cabeça, a seguinte pergunta: e se a soma der maior do que 255 (ou seja FF') que é o ***máximo*** que ***cabe*** num registro? E no caso de pares de registros, se o resultado for maior do que 65535 ('FFFF')?

De fato, em BASIC quando uma variável ***estoura*** o limite máximo do computador, o programa pára e aparece uma mensagem de erro.

Para ver o que acontece em linguagem de máquina, façamos um exemplo: vamos carregar o registrador ***BC*** com o valor máximo e somar 1. (Tabela 4).

MEM.	30000	LD E,34	'1E22'	
MEM.	30002	LD D,66	'1642'	
MEM.	30004	LD A,D	'7A'	; copia D em A
MEM.	30005	ADD A,E	'83'	; soma E com A, ou seja, com D
MEM.	30006	LD C,A	'4F'	; copia o resultado em C
MEM.	30007	LD B,0	'0600'	; coloca 0 em B para a saída do programa
MEM.	30009	RET	'C9	

Tab. 3 Programa para somar os registros D e E.

MEM.	30000	LD HL,1	'210100'	; coloca 1 em HL, ou seja, '0001' (note que devemos completar os 2 bytes de dados)
MEM.	30003	LD BC,65535	'01FFFF'	; coloca o valor máximo em BC, ou seja, 'FFFF'
MEM.	30006	ADD HL,BC	'09'	; soma BC com HL; o resultado fica em HL
MEM.	30007	LD B,H	'44'	; coloca H em B e L em C para poder ter o resultado na tela
MEM.	30008	LD C,L	'4D'	
MEM.	30009	RET	'C9'	

Tab. 4 Soma de 65535 com 1.

Você obtém como resultado o número Ø! De fato, pensando em hexadecimal (revista número *4* e número *X* de *Microhobby*), ao somar '1' ao número 'FFFF' deveríamos obter '1ØØØØ'; mas veja, este último dígito "não cabe" nos registros! Quando isto ocorre, dizemos que houve um "vai um" ou CARRY, assim como ocorre ao somarmos, por exemplo:

```
   1
  19
+ 18  (9 + 8 = 17, vai um. . .)
  --
  37
```

O microcomputador assinala este acontecimento em um bit chamado CARRY que faz parte de um registro INTERNO chamado *F* o qual *não* podemos usar DIRETAMENTE. Este registro armazena bits para várias informações (usualmente estes bits são chamados de FLAGs). Assim, sempre que executarmos uma instrução de ADD, o bit de CARRY é calculado: se houver "vai um" ele resulta *1*, caso contrário Ø.

Podemos usar o valor deste bit para fazer contas com números maiores do que 255 ou até mesmo maiores que 65536. Para isto, usaremos a instrução ADC que é abreviação de "ADD WITH CARRY" ou seja, some com o CARRY, que simplesmente adiciona ao resultado obtido de uma soma, o valor do CARRY. Teremos então as seguintes instruções: (Tabela 5).

INSTRUÇÃO		CÓDIGO
ADC	A,A	'8F'
ADC	A,B	'88'
ADC	A,C	'89'
ADC	A,D	'8A'
ADC	A,E	'8B'
ADC	A,H	'8C'
ADC	A,L	'8D'

INSTRUÇÃO		CÓDIGO
ADC	HL,BC	'ED4A'
ADC	HL,DE	'ED5A'
ADC	HL,HL	'ED6A'

Tab. 5 **A instrução ADC.**

Observação: **Estas três instruções fazem parte das instruções precedidas por 'ED'.**

Vamos então utilizar a instrução ADC para fazer contas com números maiores do que 65535. Suponhamos que quiséssemos somar os seguintes números:

com 'Ø2A5EF' (= 2*65536 + 165*256 + 239 = 173 551)
'F4DE' (= 244*256 + 222 = = 62686)
(NOTA: $65536 = (256)^2$)

Devemos obter: 173 551 + 62686 = = 236.237.

Vamos dividir os números em três partes: colocando o primeiro nos registros *B, D* e *H* e o segundo nos registros *C, E* e *O* o resultado será colocado nas memórias, após o fim do programa! O raciocínio a ser usado é análogo ao somar de números de três dígitos na base 1Ø, só que, no caso, a base seria 256. . . (Tabela 6).

Repare que não nos interessa o resultado final dos registradores *BC* que no caso será 614Ø6. O resultado da nossa *soma por partes* está nas memórias 3ØØ28, 3ØØ29 e 3ØØ3Ø que estão *após a instrução de RET*. Portanto, para verificar se o nosso raciocínio está certo, basta fazer:

PRINT 65536 * PEEK 3ØØ3Ø + 256 * PEEK 3ØØ29 + PEEK 3ØØ28

e obteremos 236.237! Note que a primeira instrução de soma deve ser *SEMPRE* um ADD pois não sabemos qual o valor inicial do CARRY e, se por acaso ele for *1*, introduziremos um erro no resultado. Neste ponto vale a pena salientar que as instruções LD *não* afetam a flag de CARRY; repare que as usamos *livremente* entre as instruções de adição sem afetar o resultado.

As instruções de ADD e ADC também podem ser utilizadas para somar constantes numéricas diretamente com o acumulador; assim temos: (Tabela 7).

INSTRUÇÃO	CÓDIGO
ADD A,dado	'C6' + 1 byte de dados
ADC A,dado	'CE' + 1 byte de dados

Tab. 7

O resultado ficará no acumulador; lembre-se que este registro é ***privilegiado*** e todas as operações aritméticas são referidas a ele! Obviamente estas instruções também afetam a flag de CARRY. Note que, não podemos somar dados DIRETAMENTE com pares de registros!

Suponhamos então que você deseje somar um número, menor que 256 (por exemplo 74) com um par de registros (por exemplo *HL*). Isto poderia ser feita da seguinte maneira: (Tabela 8)

Repare cuidadosamente no uso da instrução ADC A,Ø, que foi utilizada apenas para somar o valor do CARRY ao registro H! Usando este mesmo procedimento, tente agora somar um número maior do que 255 ao par HL, por exemplo HL + + 256. . . (você deverá ***quebrar*** o número em dois bytes!). O exemplo anterior poderia também ser feito carregando o par DE com 74 e usando a instrução ADD HL, DE; entretanto, estaremos usando mais registros.

A operação de ***soma*** é muito utilizada também quando desejamos utilizar uma variável como ***contador*** para poder fazer ***loops*** de ***repetição***; mais adiante, aprenderemos como fazer esses ***loops*** em linguagem de máquina; por enquanto, vamos introduzir a instrução INC (abreviação de INCREMENT), a qual adiciona 1 (um) ao valor do determinado registro (ou par), SEM, no entanto, AFETAR O VALOR DO CARRY! Observe que podemos usar →

MEM.	3ØØØØ	LD E,2ØØ	'1EC8'	
MEM.	3ØØØ2	LD D,58	'163A'	
MEM.	3ØØØ4	LD A,D	'7A'	
MEM.	3ØØØ5	ADD A,E	'83'	; soma E com A e gera CARRY
MEM.	3ØØØ6	LD C,A	'4F'	; coloca o resultado em C
MEM.	3ØØØ7	LD A,Ø	'3EØØ'	; coloca Ø em A
MEM.	3ØØØ9	ADC A,A	'8F'	; transfere o CARRY para A e a seguir para B
MEM.	3ØØ1Ø	LD B,A	'47'	
MEM.	3ØØ11	RET	'C9'	

Tab. 6 Soma de dois registros com resultado maior que 255.

MEM.	3ØØØØ	LD DE,13189	'118533'	; coloca 13189 em DE
MEM.	3ØØØ3	LD HL,31687	'21C77B'	; coloca 31687 em HL
MEM.	3ØØØ6	ADD HL,DE	'19'	; soma DE com HL
MEM.	3ØØØ7	LD B,H	'44'	; coloca H em B e L em C para a saída na tela
MEM.	3ØØØ8	LD C,L	'4D'	
MEM.	3ØØØ9	RET	'C9'	

Tab. 8 Soma dos pares DE com HL.

esta instrução com TODOS os registros: (Tabela 9).

INSTRUÇÃO	CÓDIGO
INC A	'3C'
INC B	'Ø4'
INC C	'ØC'
INC D	'14'
INC E	'1C'
INC H	'24'
INC L	'2C'
INC BC	'Ø3'
INC DE	'13'
INC HL	'23'

Tab. 9

Novamente, referindo-se ao BASIC, uma analogia para esta instrução seria:

LET X = X + 1

Lembre-se sempre da diferença entre ADD e INC: por exemplo, ADD A,1 irá calcular um valor para o CARRY (Ø ou 1) enquanto que INC A deixará o CARRY inalterado, embora as duas instruções produzam o mesmo resultado no acumulador.

Vamos então, seguindo os mesmos passos que fizemos com a instrução LD, sair dos registros e ir para a memória. Apenas três instruções são disponíveis, que mostram claramente o ***privilégio*** do Acumulador e do par de registros HL: (Tabela 10).

INSTRUÇÃO	CÓDIGO
ADD A,(HL)	'86'
ADC A,(HL)	'8E'
INC (HL)	'34'

Tab. 10 As instruções INC e ADD usando a memória.

que significam respectivamente: some o CONTEÚDO da memória indicado pelo par HL ao acumulador (sem CARRY e com CARRY) e incremente o conteúdo da memória indicada por HL. Assim, vejamos o procedimento necessário, para somar um número, por exemplo, 64, ao conteúdo da memória 3ØØ1Ø: (Tabela 11).

Faça POKE 3ØØ1Ø,36 e a seguir execute o programa. Você deverá obter 1ØØ, pois irá realizar a conta 64 + 36 = 1ØØ! É sempre bom recordar que as variáveis devem ficar APÓS a instrução de RET (3ØØ1Ø) no nosso caso). Entretanto, para programas mais complexos, é interessante colocar as variáveis ***bem depois*** da instrução de RET, deixando espaço livre na memória entre o fim do programa e as variáveis para eventuais modificações no programa que possam causar ***aumento*** do seu tamanho. ○

MEM.	3ØØØØ	LD	HL,3ØØ1Ø	'213A75'	; coloca 3ØØ1Ø em HL
MEM.	3ØØØ3	LD	A,64	'3E4Ø'	; coloca 64 em A
MEM.	3ØØØ5	ADD	A,(HL)	'8E'	; soma o conteúdo da memória 3ØØ1Ø com A
MEM.	3ØØØ6	LD	B,Ø	'Ø6ØØ'	; transfere o resultado para BC
MEM.	3ØØØ8	LD	C,A	'4F'	
MEM.	3ØØØ9	RET		'C9'	

Tab. 11 A instrução INC para os registros internos.

As 7 Pontes no Caminho da Aldeia

Renato da Silva Oliveira

Novamente, para esta seção, contamos com a colaboração de nosso colega, assinante e amigo pessoal, Nabor Rosenthal.

Recebemos por volta do começo do mês, uma carta onde Nabor nos conta sobre sua última viagem, quando esteve no Norte da Índia.

Lá, a caminho da casa de seu amigo, Ramarujan, ele teve que atravessar uma intrincada rede de pontes sobre um rio. O passeio de Nabor Rosenthal foi o que nos inspirou este mês.

A seguir, transcreveremos o trecho da carta que deu origem ao nosso Quebra-Cabeça:

"Indo ao encontro de meu amigo de infância, tive que atravessar um rio. Este rio continha 10 pequenas ilhas, interligadas por uma rede de 18 pontes. Entretanto, para cruzar sete destas pontes, deveríamos pagar uma determinada quantia. Este fato chamou-me a atenção e perguntei-me: Por que apenas sete?

Eu queria visitar todas as ilhotas, pois uma carta de meu amigo hindú me informava que em cada uma delas havia uma escultura em pedra, verdadeiras obras de arte, datadas de muitos séculos atrás.

Antes de atravessar, porém, eu pensei um pouco sobre as sete pontes. Não foi preciso muito tempo para descobrir a milenar sabedoria Hindu, escondida discretamente na disposição das pontes e na escolha das pontes que deveriam ser taxadas. Estas sete pontes foram escolhidas de tal forma que quem quisesse se livrar das taxas, teria que passar necessariamente por todas as ilhotas e por todas as outras 11 pontes, pelo menos uma vez. Desse modo, uma pessoa disposta a não gastar desnecessariamente, veria todas as dez esculturas, sem pagar um centavo.

Atravessei o rio pelo caminho mais econômico (sem pagar nada), passando uma única vez por cada ilhota e por cada uma das 11 pontes sem taxas. Na base de cada escultura que encontrei em meu caminho, havia um número gravado. Anotei cada um deles e vi que formavam uma sequência lógica que chegou a me espantar, pois estes números, nesta sequência, aparentemente apresentavam um significado".

Em anexo, Nabor nos mandou um mapa, que reproduzimos na figura 1. As pontes estão diferenciadas por letras e as ilhas pelos números das esculturas.

— Gatinho Careteiro! — disse ela com timidez, não sabendo se o Gato gostava que o chamassem assim. Vendo que não se zangava, aventurou-se a concluir a frase:

— Pode dizer-me que caminho devo tomar?

— Isso depende do lugar para onde quer ir — respondeu com muito propósito o Gato.

— Não tenho destino certo.

— Nesse caso, qualquer caminho serve.

— Servirá sim, se o caminho levar a algum lugar — sugeriu Alice.

— Qualquer caminho conduz a algum ponto, se você andar depressa e chegar — disse o Gato.

Trecho de "Alice no País das Maravilhas" de Lewis Carroll — Adaptado por M. Lobato

Seu problema é:

1) Fazer com que o TK, previamente programado, descubra o caminho seguido por Nabor Rosenthal, indicando quais das pontes ele usou.

2) Descobrir, usando o TK, se há algum significado na sequência de números encontrados nas bases das esculturas por Nabor Rosenthal, e se houver, esclarecê-lo. O

Aqui você tem a melhor iniciação em microcomputação que existe.

O TK 83 já ensinou mais de 2 milhões de pessoas. Ele é muito fácil de operar. Usa o Basic, e a memória chega até 64 K bytes, e aceita monitor, impressora e joystick.

Num instante você vai estar resolvendo problemas programando, ou vencendo os muitos jogos disponíveis. O TK 83 não é só a melhor iniciação. Também é a mais divertida.

Aqui você já aplica os seus conhecimentos

Com o TK 85 você também pode se divertir muito: ele tem dezenas de jogos disponíveis.

Mas ele já é mais sofisticado. Tem software já pronto. Linguagens Basic e Assembler. Teclado tipo máquina de escrever, com 40 teclas e 160 funções. 16 ou 48 K de memória RAM, e 10 de ROM. Gravação em high-speed, e função Verify, para maior segurança.

Quando você já estiver apaixonado por microcomputação, ele vai corresponder totalmente.

Aqui você mostra tudo o que sabe.

O TK 2000 Color tem tudo que os melhores micros têm. Menos o preço. Aceita diskette, impressora (já vem com interface), alta resolução gráfica à cores podendo ser ligado ao seu TV colorido ou P&B. Tem 64 k de memória RAM e 16 k de memória ROM. Com excelente software disponível.

Você pode mostrar tudo o que sabe. Sem precisar mostrar muito dinheiro.

* Preço sujeito a alteração

MICRODIGITAL

Microdigital Eletrônica Ltda
Caixa Postal 54121 - CEP 01000 -
São Paulo - SP Telex nº (011) 37008 MIDE BR

À venda nas boas casas do ramo, lojas especializadas de fotovideo-som e grandes magazines em: **ALAGOAS** - Maceió, Palmeira dos Indios, **AMAZONAS** - Manaus, **BAHIA** - Salvador, **CEARÁ** - Fortaleza, **DISTRITO FEDERAL** - Brasília, **ESPÍRITO SANTO** - Vitória, **GOIÁS** - Goiânia, **MATO GROSSO** - Cuiabá, **MINAS GERAIS** - Belo Horizonte, Divinópolis, Itajuba, Juiz de Fora, Poços de Caldas, São João Del Rei, Teófilo Otoni, Uberlândia, Uberaba, Viçosa, **PARAÍBA** - Campina Grande, **PARÁ** - Belém, **PARANÁ** - Curitiba, Londrina, Maringá, **PERNAMBUCO** - Recife, **RIO DE JANEIRO** - Campos, Niterói, Nova Friburgo, Petrópolis, Rezende, Rio de Janeiro, Volta Redonda, **RIO GRANDE DO SUL** - Bagé, Canoas, Caxias do Sul, Ijui, Novo Hamburgo, Pelotas, Porto Alegre, Sant'Anna do Livramento, Santiago, Santa Rosa, São Leopoldo, **RIO GRANDE DO NORTE** - Natal, **RONDÔNIA** - Porto Velho, **SÃO PAULO** - Araraquara, Assis, Avaré, Bauru, Birigui, Botucatu, Campinas, Catanduva, Franca, Guarulhos, Itu, Jacarei, Jaú, Limeira, Lins, Marilia, Mogi Guaçu, Mogi das Cruzes, Ourinhos, Piracicaba, Pirassununga, Promissão, Rio Claro, Ribeirão Preto, Santos, Santa Barb. d'Oeste, São Bernardo do Campo, São João da Boa Vista, São Sebs. da Grama, São Carlos, São José do Rio Preto, São José dos Campos, Stº André, São Paulo, Sorocaba, Suzano, Taubaté, **SANTA CATARINA** - Blumenau, Brusque, Florinópolis, Itajaí, Joinville, **SERGIPE** - Aracajú. Se você não encontrar este equipamento na sua cidade ligue para (011) 800 - 255.8583.